９．割り込みを学ぼう

回路製作の詳細は第０章を参照してください．

このソースファイルを各自打ち込んで下さい.

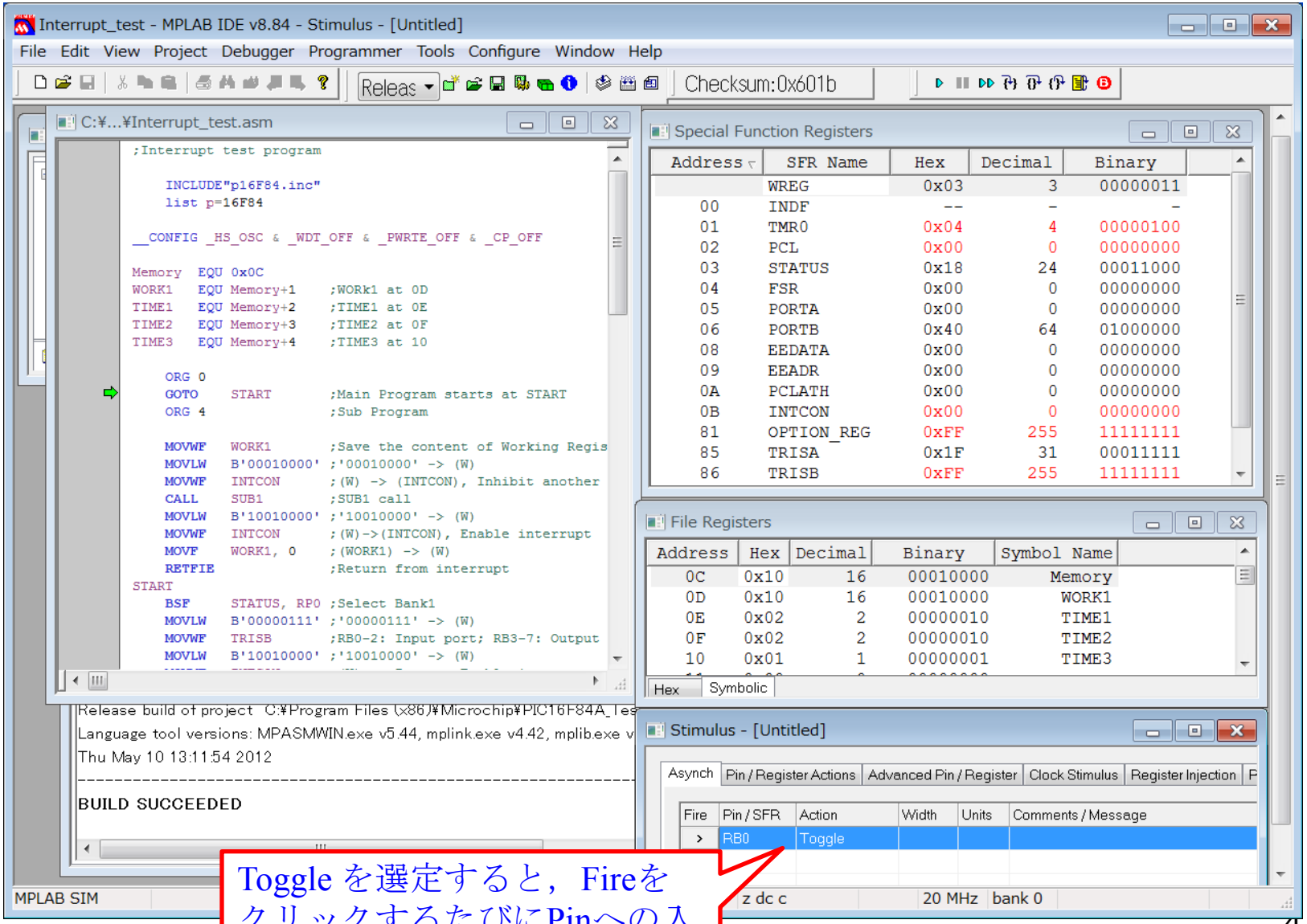
```
;Interrupt test program

            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
```

ソースファイル（続き）

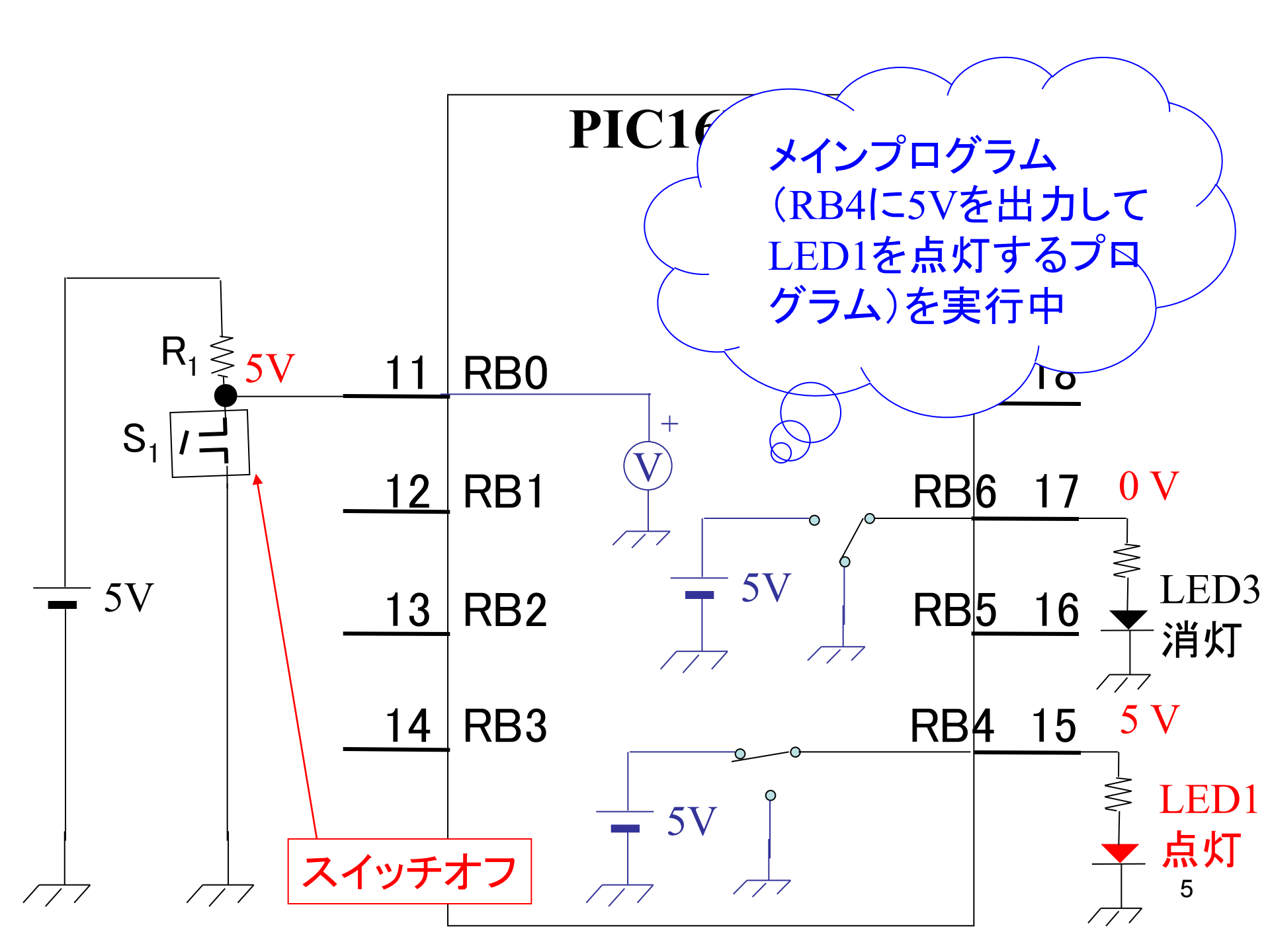
```
;Main Program

            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
```

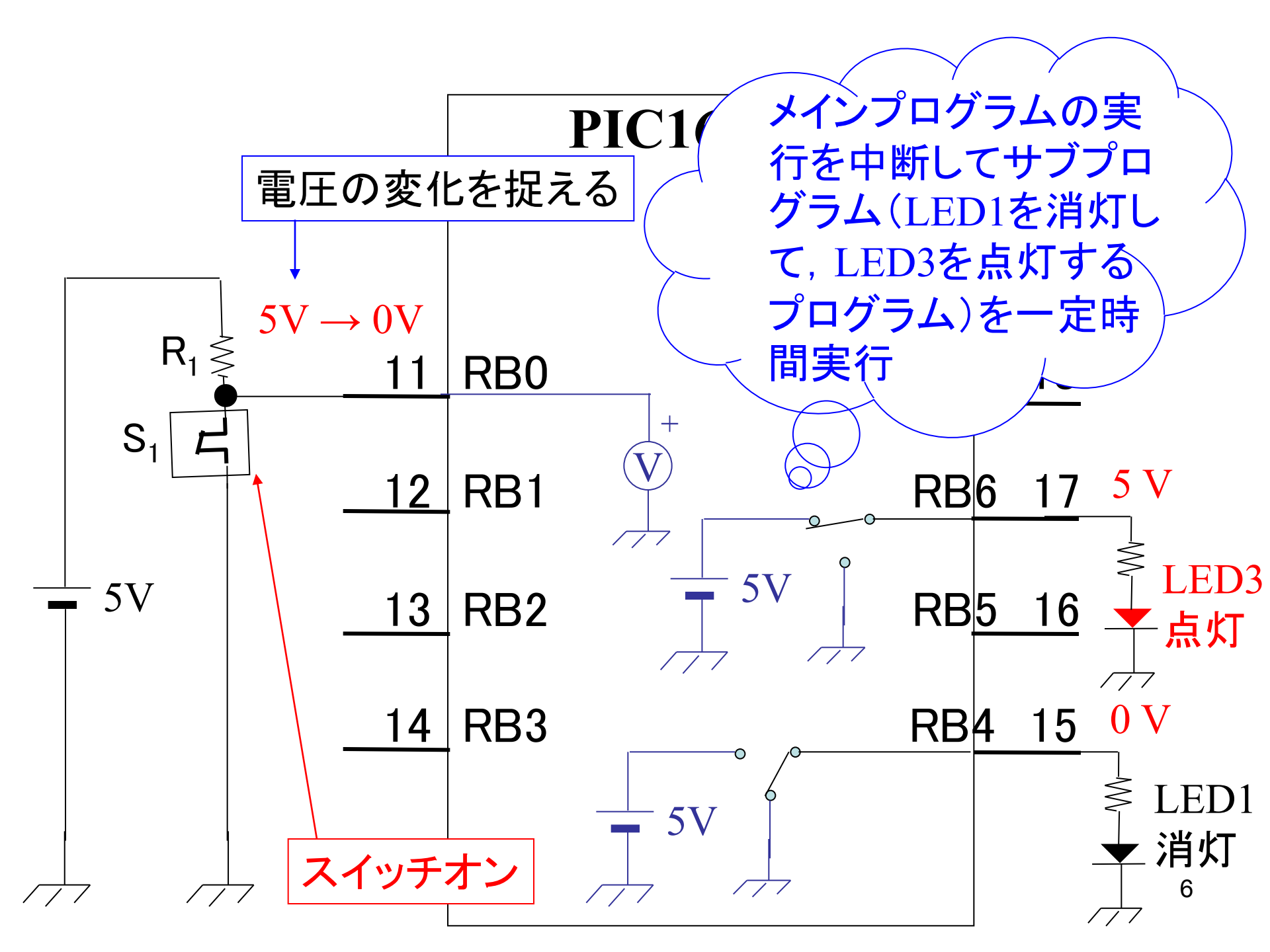
デモプログラムでは80となっていますが，シミュレーションではくり返し回数が多すぎるので0x03を入力してみて下さい．

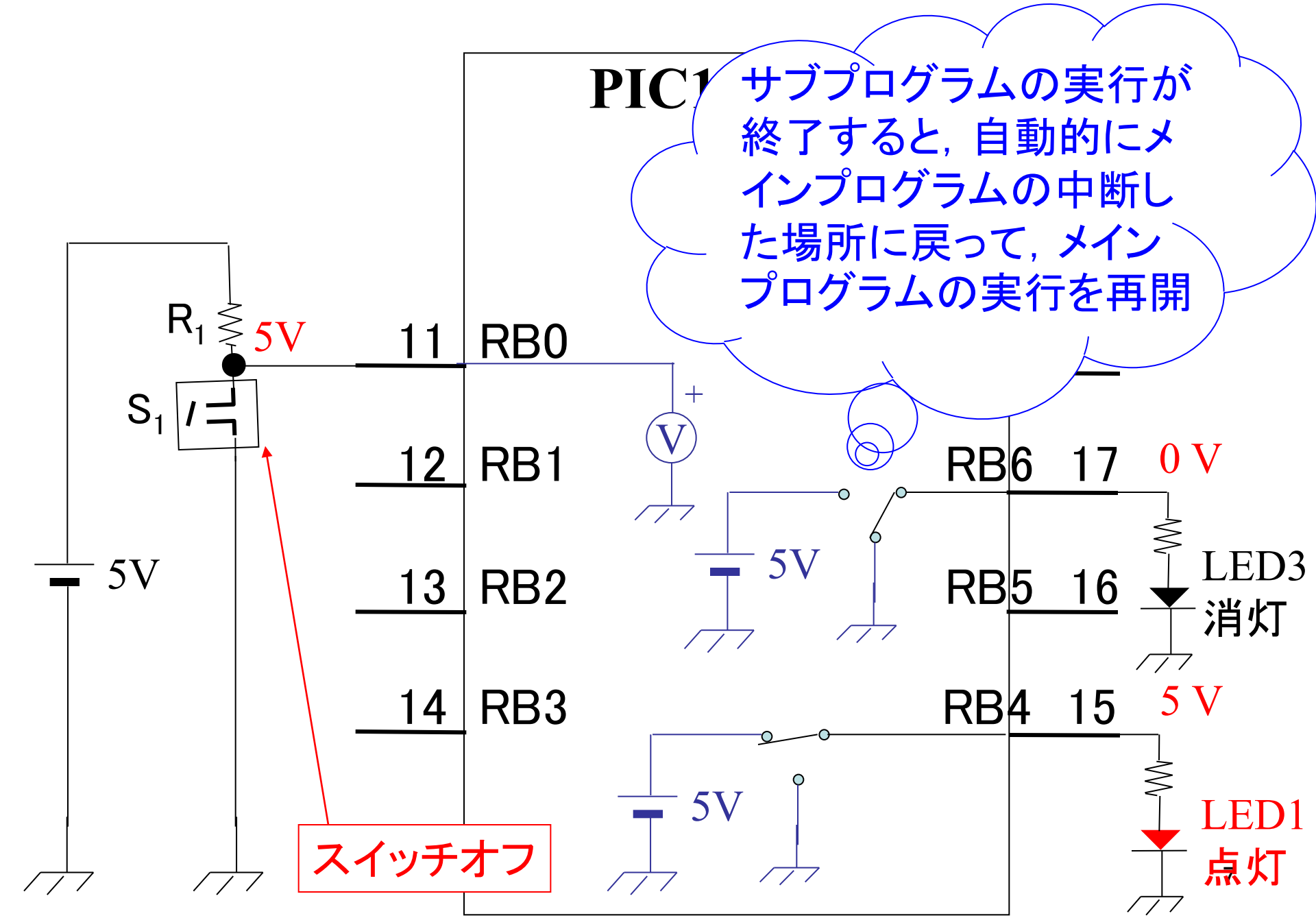
Debugger → Select Tool → MPLAB SIM → Make → View → Special Function Registers → View → File Registers → Stimulus → New Workbook → RB0+Toggle → F7 を押し続けながら ときどき RB0をFire → ・・・



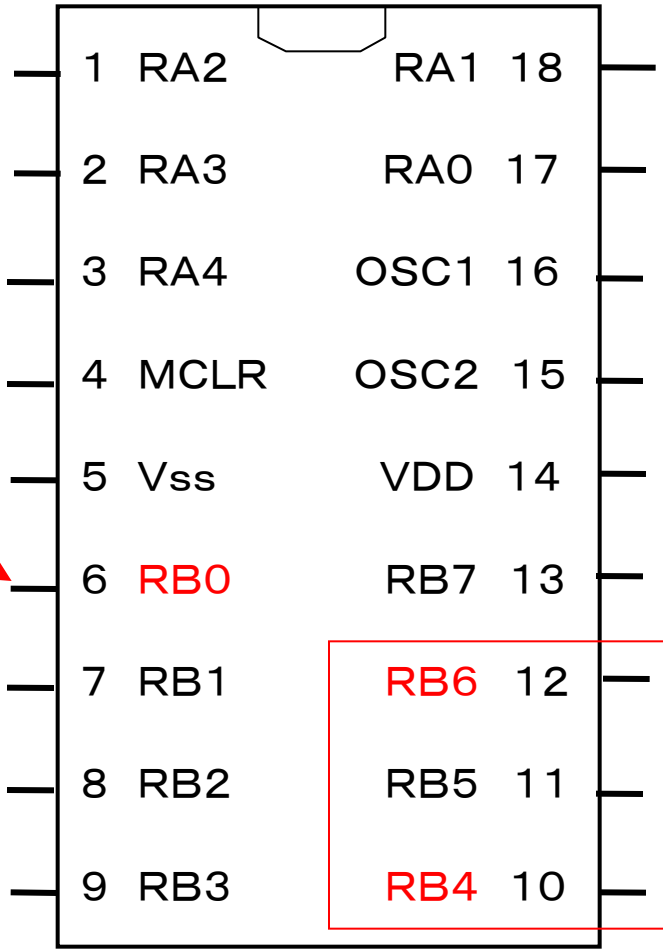
Toggle を選定すると，Fireをクリックするたびにに Pinへの入力が反転する．


PIC16
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）を実行中
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
5V
5V
5V
V
+
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を消灯して，LED3を点灯するプログラム）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
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

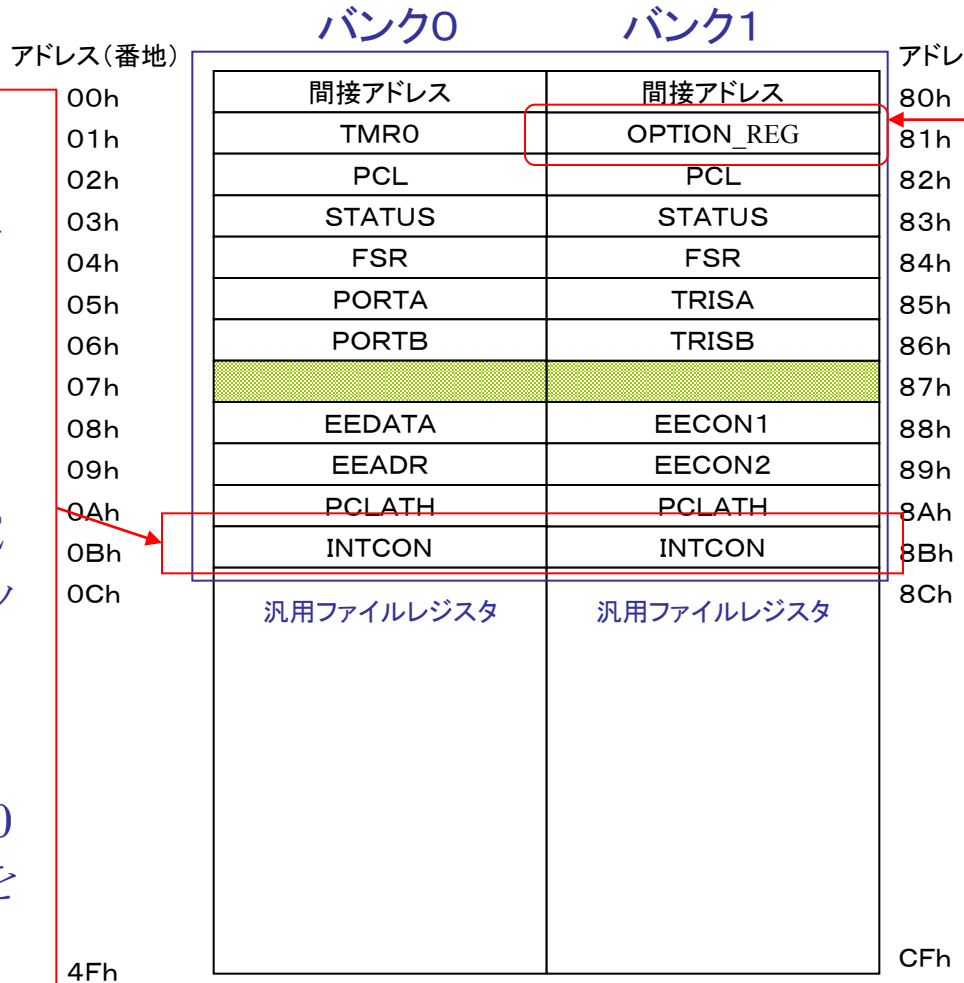
LEDを点灯させる.

本章のポイント

INTCONレジスタ
p.119

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
p.119

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．

図１．１６　ファイルレジスタの配置，p. 15

表1.2 特殊レジスター一覧(p.16)

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
            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
```

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(p.12)に格納され，プログラムカウンタ(p.12)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）を実行すると，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.

```
;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(p.122)

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(p.119)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

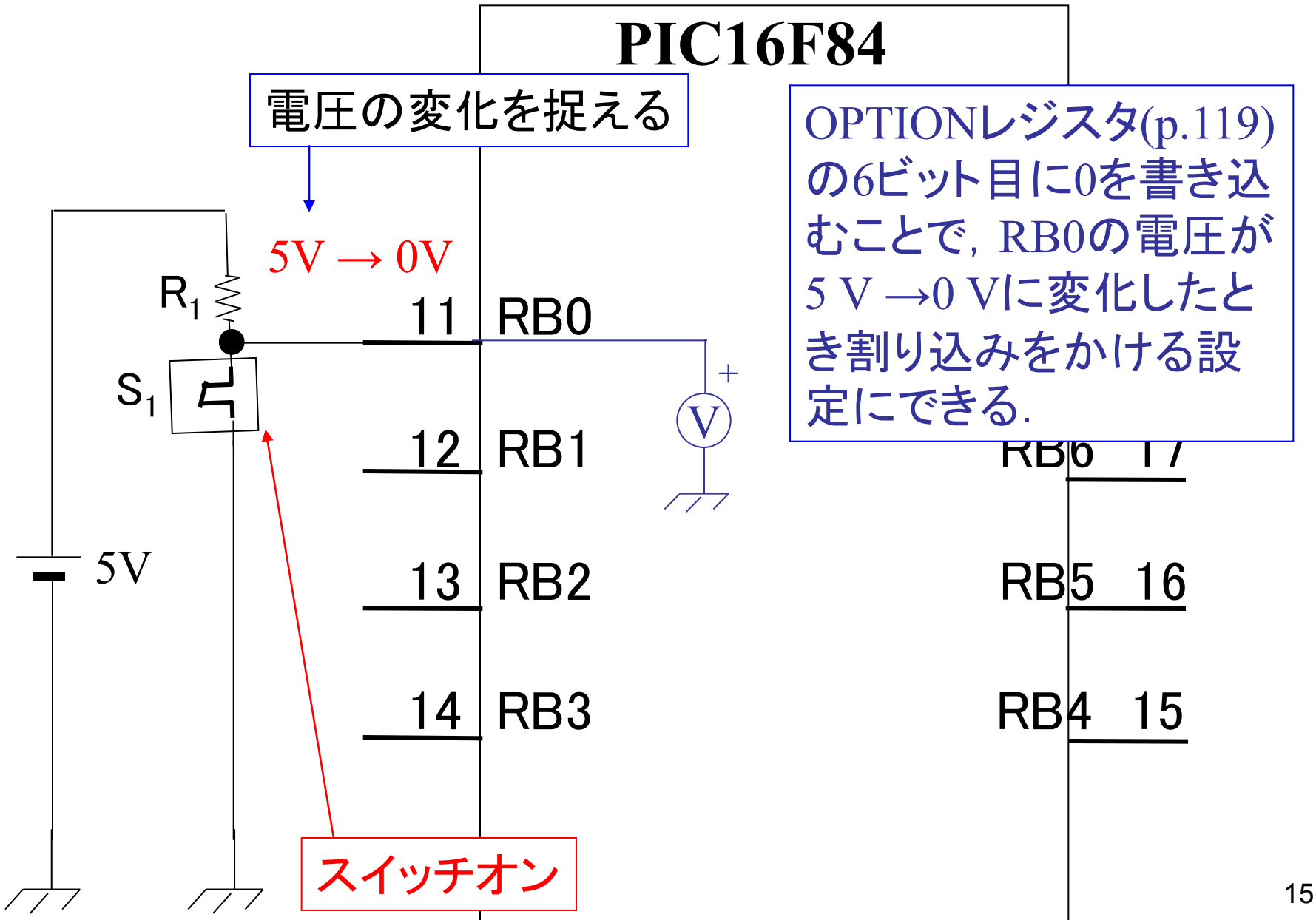
```
;Interrupt test program

            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(p.119)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(p.119)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


;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
```

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(p.12)に格納され，プログラムカウンタ(p.12)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.

```
;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
```

割り込み時のメインプログラムのデータの退避場所.

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.

```
;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
```

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.

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

LED1を消灯し，LED3を点灯する信号をPORTBに出力する.

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．

```
;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
```

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．

演習問題21. メインプログラムにおいてW, MEM1レジスタを使うプログラムを書け. そして, 割り込み処理が発生した場合に, これらW, MEM1レジスタをWORK1, WORK2に退避するプログラムを書け.

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