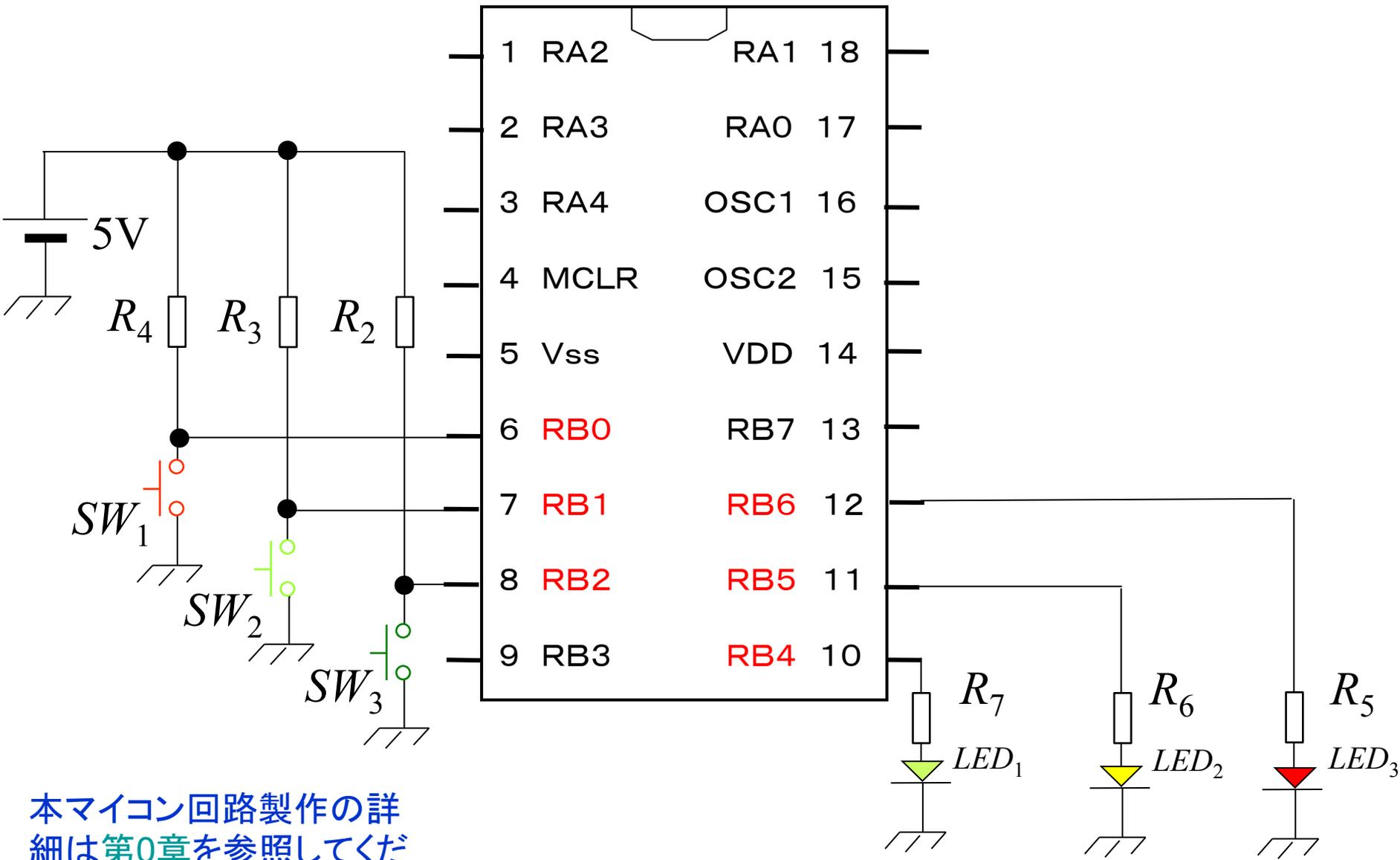


6. LED (発光ダイオード) の制御を学ぼう

本稿のWebページ

<http://www.mybook-pub-site.sakura.ne.jp/PIC/index.html>

PIC16F84A



本マイコン回路製作の詳細は第0章を参照してください。

; LED control program

```
INCLUDE "p16F84A.inc"  
list p=16F84A
```

```
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
MEM1 EQU 0x0C
```

```
ORG 0  
GOTO START  
ORG 4
```

```
START
```

```
BSF STATUS,RP0 ;Port B Setting  
;Selection of Bank 1  
MOVLW B'00000111'  
MOVWF TRISB ;RB0-2 -> Input Port, RB3-7 -> Output Port  
BCF STATUS,RP0 ;Selection of Bank 0
```

```
STEP1
```

```
MOVF PORTB,0 ;(RB)->(W)  
ANDLW B'00000111' ;(W) and 00000111 -> (W)  
MOVWF MEM1 ;(W) -> (MEM1)  
RLF MEM1,1 ;Rotate Left(MEM1) -> (MEM1)  
MOVF MEM1,0 ;(MEM1) -> (W)  
MOVWF PORTB ;Output to Port B  
GOTO STEP1 ;Repetition of Processing
```

```
END
```

このソースファイルを打ち込んで下さい。
シミュレータで動作確認が終わったら、PICマイコンにプログラムの書き込みを行い、動作確認をして下さい。

これはスイッチを閉じるとLEDが消灯し、スイッチを開くとLEDが点灯するプログラムです。

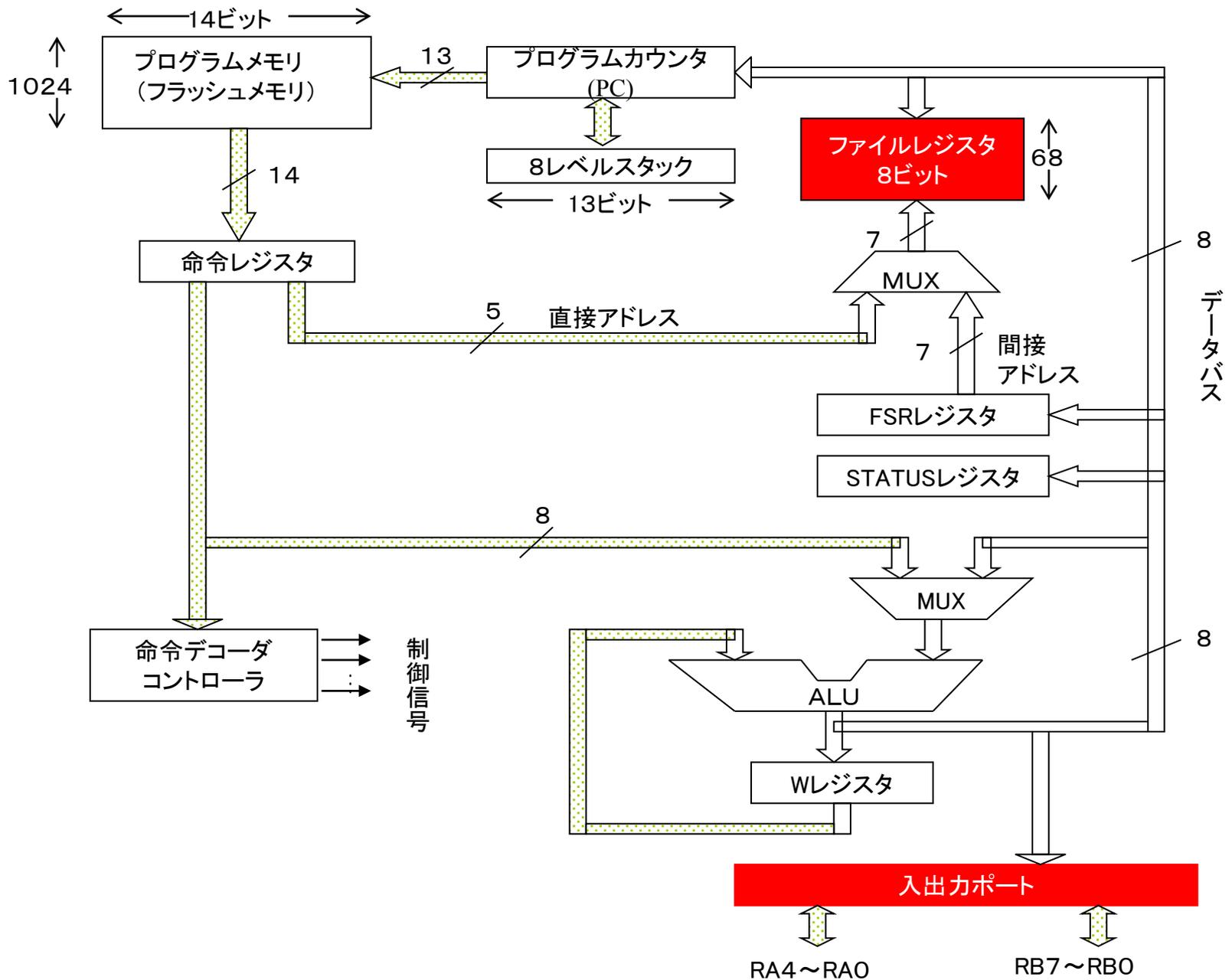
SW₁, SW₂, SW₃ の入力をシミュレートするにはDebugger → Stimulus → New Workbook
 としてPin/SFRの列のセルを左クリックしてプルダウンメニューよりRB0, RB1, RB2
 を選ぶ. それぞれのActionを例えばSet High, Set Low, Set Highと設定し, それぞれの
 行の先頭のFireボタンを押す. F7ボタンを押しながらステップ実行をすると
 MOVF PORTB, 0の命令を実行した段階で, Wレジスタに'00000101'が入力される.
 (詳細は[第0章](#)を参照してください.)

The screenshot displays the MPLAB IDE interface for the LED control program. The main window shows the assembly code for LED_cont.asm, with the instruction `MOVF PORTB, 0` highlighted by a red arrow. The Special Function Registers window shows the WREG register at address 00 with a binary value of 00000101, also indicated by a red arrow. The File Registers window shows the INTCON register at address 0B. The Stimulus window shows a table of actions for pins RB0, RB1, and RB2, with the RB2 Set High action highlighted by a red arrow.

Address	SFR Name	Hex	Decimal	Binary
00	WREG	0x05	5	00000101
01	INDF	--	--	--
02	TMR0	0x00	0	00000000
03	PCL	0x09	9	00001001
04	STATUS	0x18	24	00011000
05	FSR	0x00	0	00000000
06	PORTA	0x00	0	00000000
07	PORTB	0x05	5	00000101
08	EEDATA	0x00	0	00000000
09	EEADR	0x00	0	00000000
0A	PCLATH	0x00	0	00000000
0B	INTCON	0x00	0	00000000
81	OPTION_REG	0xFF	255	11111111
85	TRISA	0xFF	31	00111111
86	TRISB	0x07	7	00000111
88	ECON1	0x00	0	00000000

Address	Hex	Decimal	Binary	Symbol Name
0A	0x00	0	00000000	PCLATH
0B	0x00	0	00000000	INTCON
0C	0x00	0	00000000	MEM1
0D	0x00	0	00000000	

Fire	Pin / SFR	Action	Width	Units	Comments / Message
>	RB0	Set High			
>	RB1	Set Low			
>	RB2	Set High			



PIC16F84Aの構成

Data Sheet の REGISTER FILE MAP より

バンク0		バンク1	
00h	間接アドレス	間接アドレス	80h
01h	TMRO	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	汎用ファイルレジスタ	汎用ファイルレジスタ	
4Fh			CFh

本章のポイント

入出力レジスタ

PORTA

PORTB

の利用です。

; LED control program

```
INCLUDE "p16F84A.inc"  
list p=16F84A
```

```
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
MEM1 EQU 0x0C
```

```
ORG 0
```

```
GOTO START
```

```
ORG 4
```

```
START
```

```
BSF STATUS, RP0
```

```
MOVLW B'00000111'
```

```
MOVWF TRISB
```

```
BCF STATUS, RP0
```

```
STEP1
```

```
MOVF PORTB, 0
```

```
ANDLW B'00000111'
```

```
MOVWF MEM1
```

```
RLF MEM1, 1
```

```
MOVF MEM1, 0
```

```
MOVWF PORTB
```

```
GOTO STEP1
```

```
END
```

ファイルレジスタのアドレスは、バンクという概念で管理されています。

RP0 (STATUSレジスタの第5ビット)を1とすることで **バンク 1** を選択できます。

BSF f, b Bit Set f の略です。
ファイル f のビット b を 1 にします。

STATUSレジスタの各ビットには、Data SheetのSTATUS REGISTERによると

IRP	RP1	RP0	\overline{TO}	PD	Z	DC	C
-----	-----	------------	-----------------	----	---	----	---

と名前がつけられています。

; LED control program

```
INCLUDE "p16F84A.inc"  
list p=16F84A
```

```
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
MEM1 EQU 0x0C
```

```
ORG 0  
GOTO START  
ORG 4
```

```
START
```

```
BSF STATUS,RP0  
MOVLW B'00000111'  
MOVWF TRISB  
BCF STATUS,RP0
```

```
STEP1
```

```
MOVF PORTB,0  
ANDLW B'00000111'  
MOVWF MEM1  
RLF MEM1,1  
RLF MEM1,1  
RLF MEM1,1  
RLF MEM1,1  
MOVF MEM1,0  
MOVWF PORTB  
GOTO STEP1
```

```
END
```

バンク 1 にある **TRISB** レジスタに

'00000111'

を書き込むと、入出力用の **PORTB** を

RB7 → 出力ポート

RB6 → 出力ポート

RB5 → 出力ポート

RB4 → 出力ポート

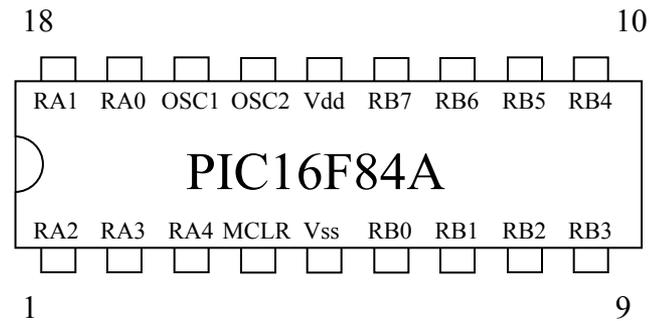
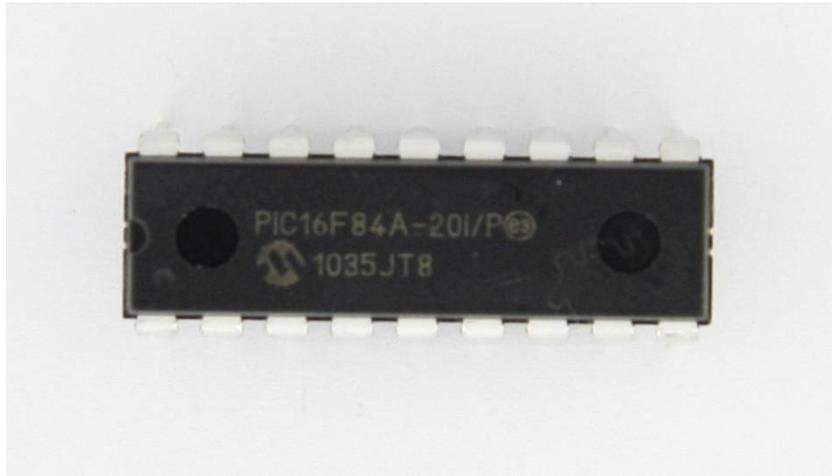
RB3 → 出力ポート

RB2 → 入力ポート

RB1 → 入力ポート

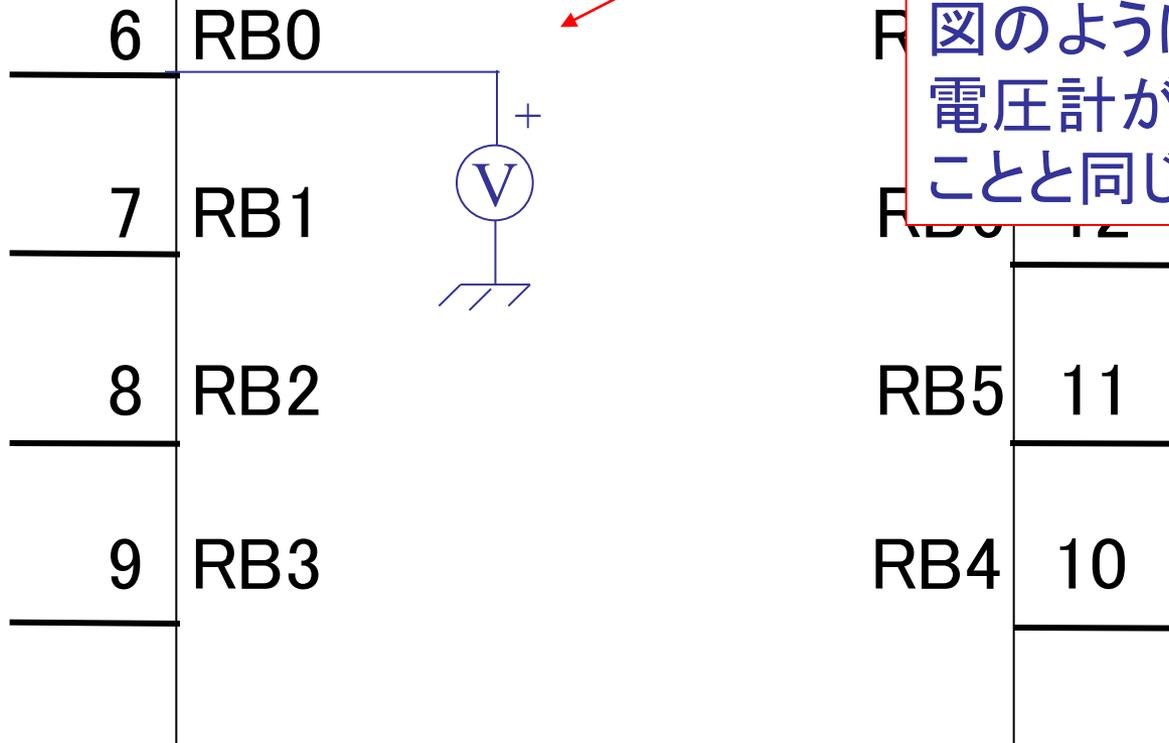
RB0 → 入力ポート

と設定できます。

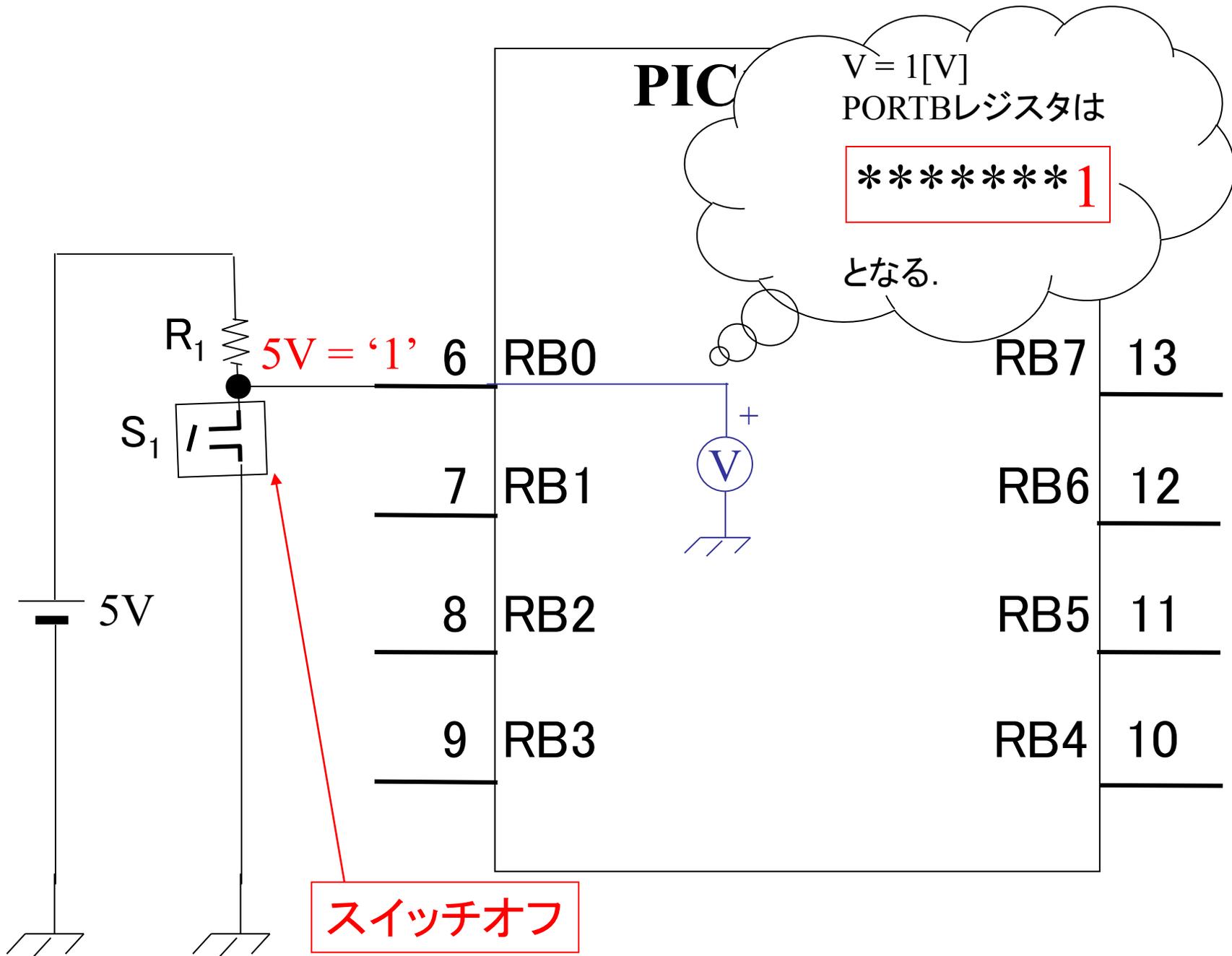


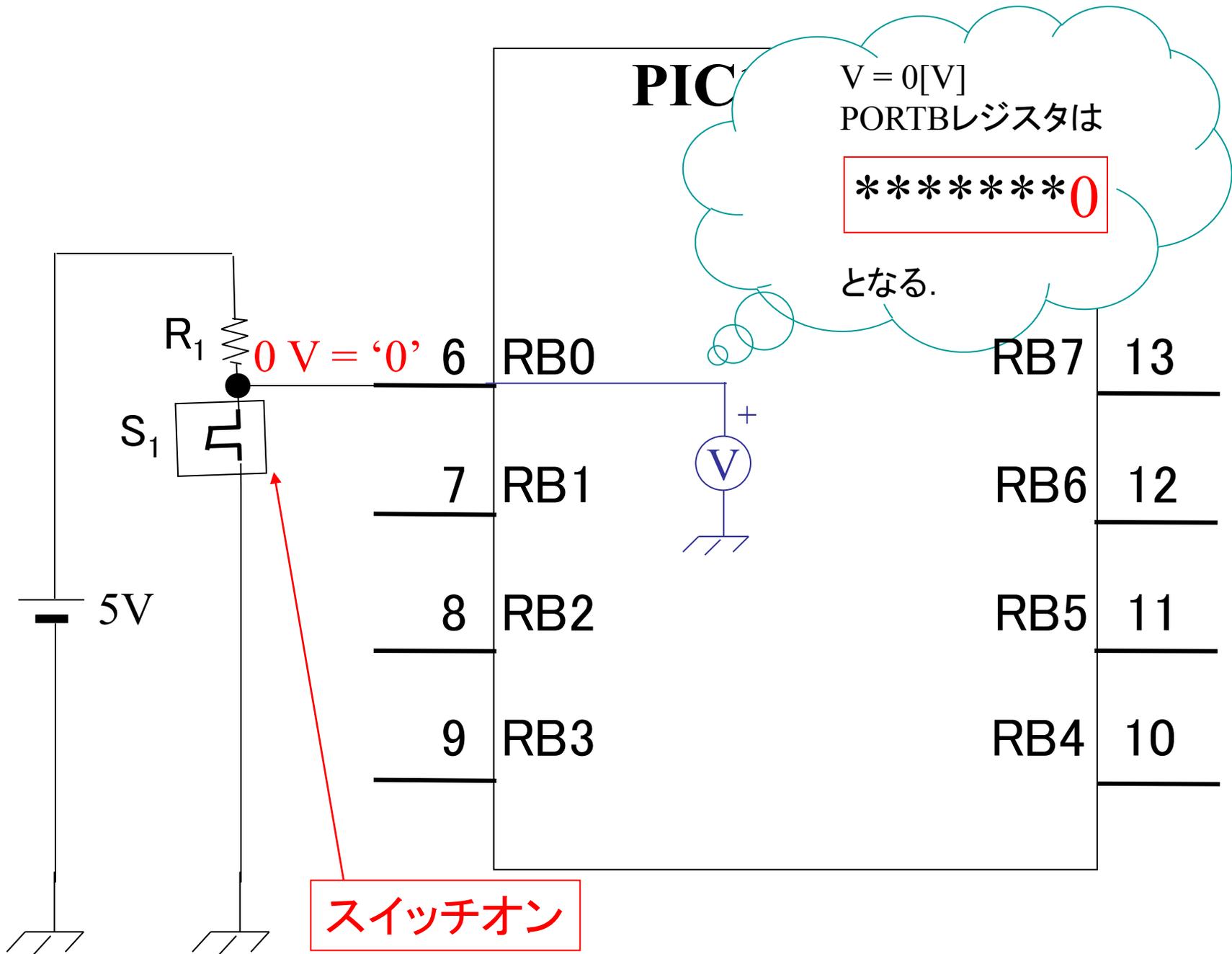
PIC16F84Aの外観とピン配置

PIC16F84A

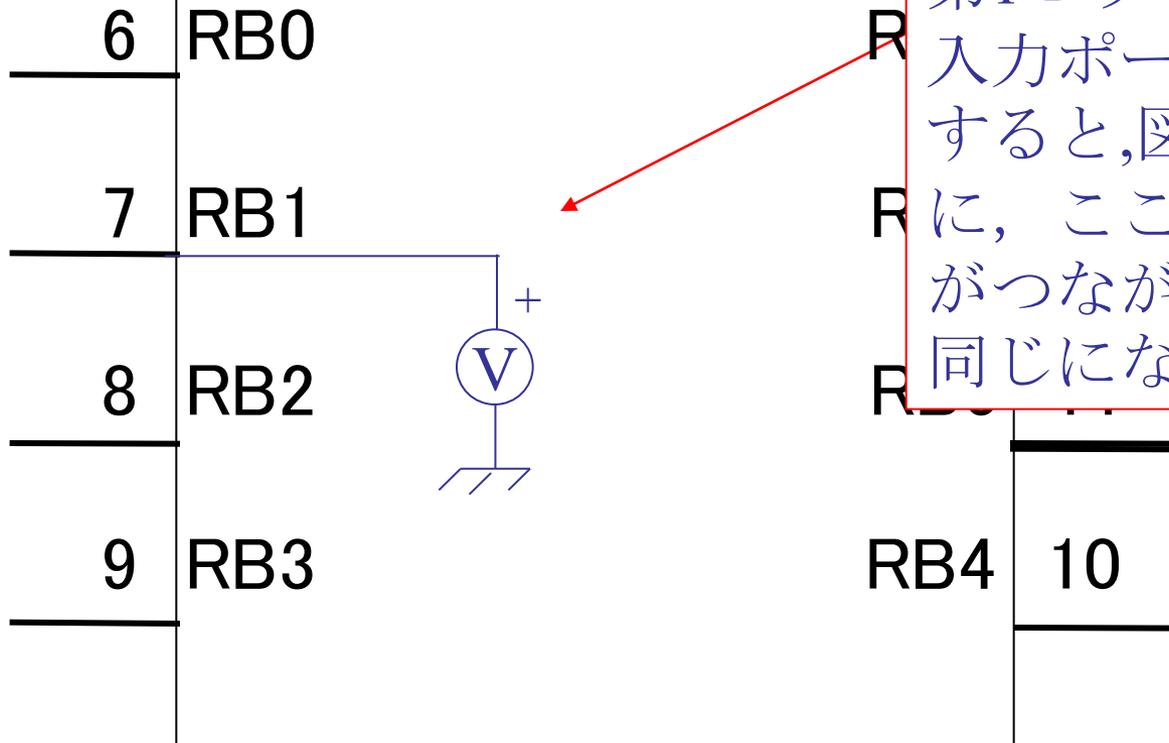


例えば, PORTBの第0ビットRB0を入力ポートに指定すると, 図のように, ここに電圧計がつながることと同じになる.

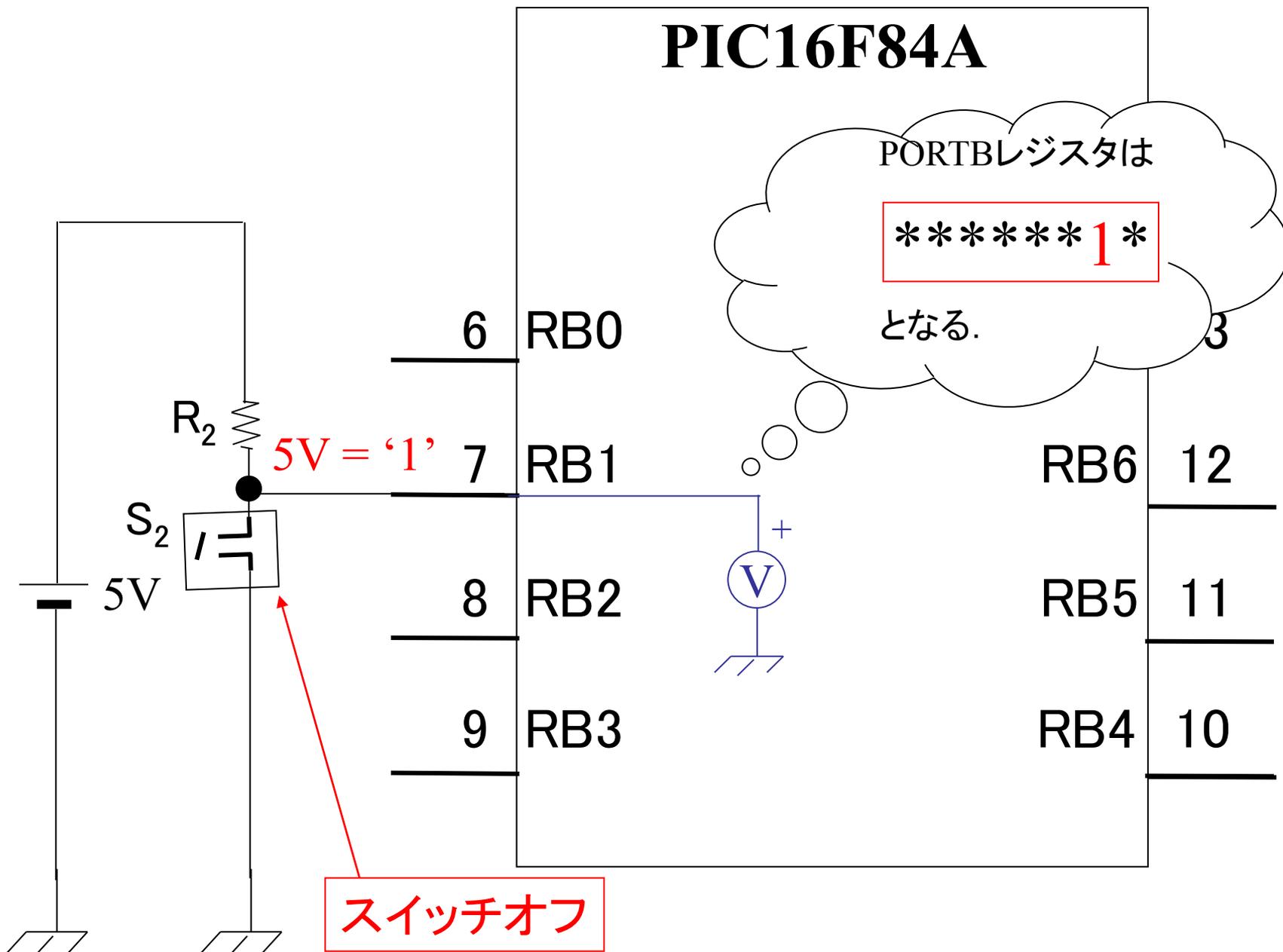


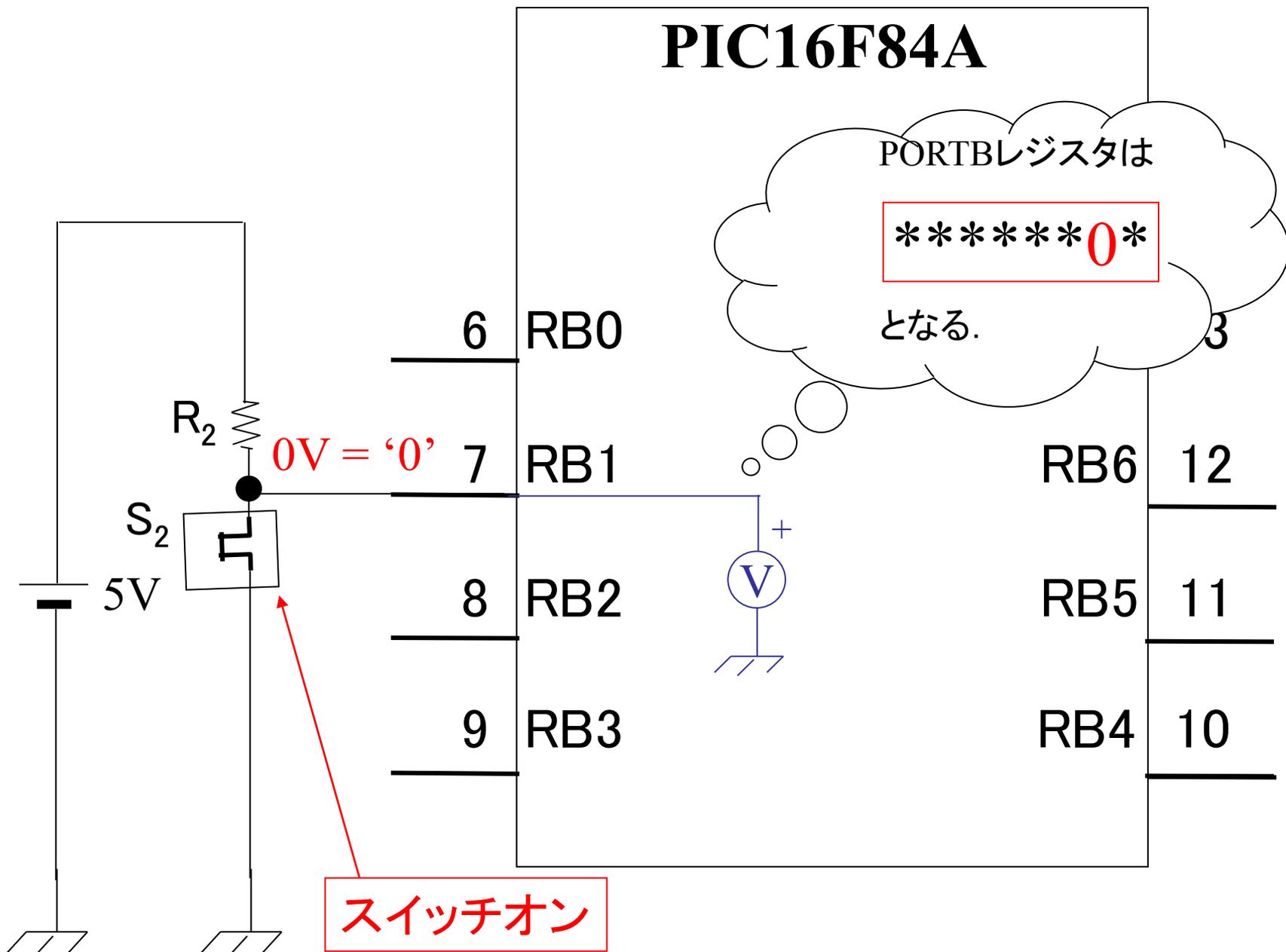


PIC16F84A



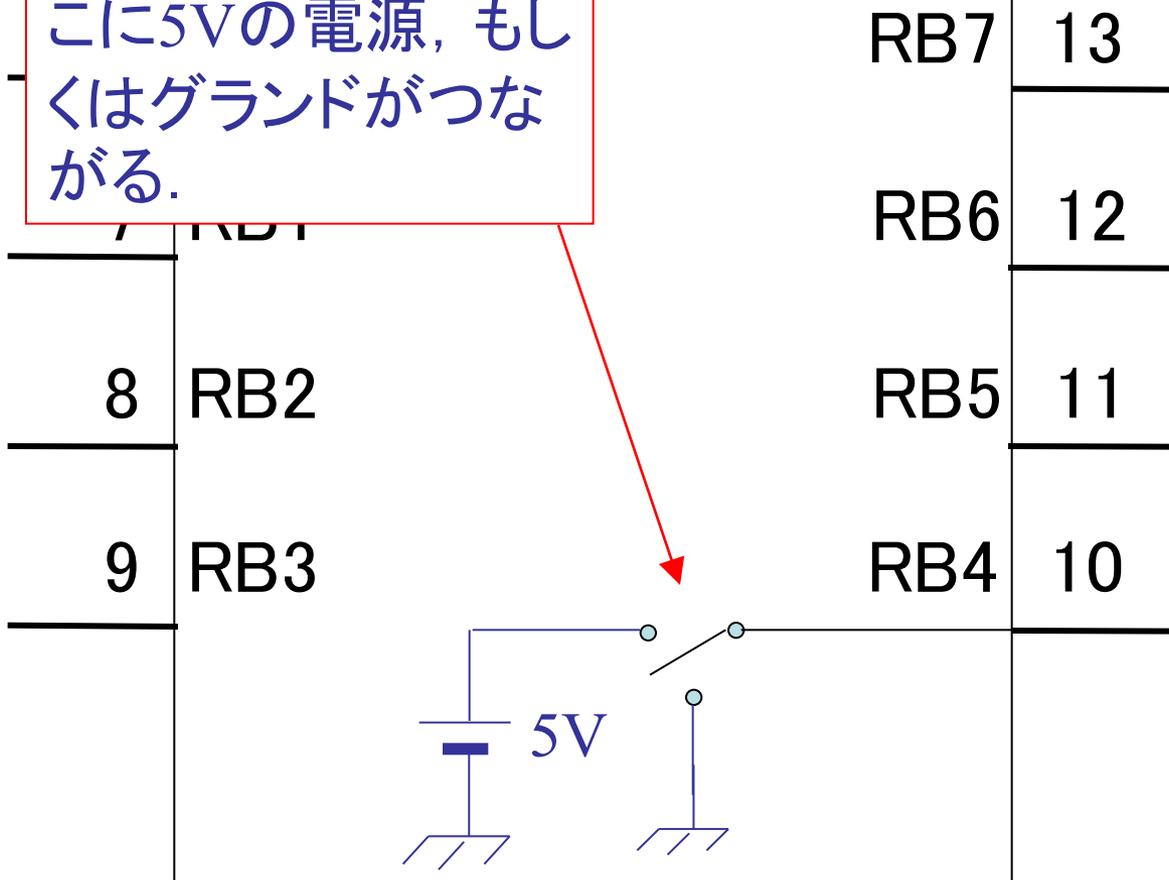
同様に，PORTBの第1ビットRB1を入力ポートに指定すると，図のように，ここに電圧計がつながることと同じになる。





PIC16F84A

例えば, PORTBの
第4ビットRB4を出
カポートに指定す
ると, 図のように, こ
こに5Vの電源, もし
くはグランドがつな
がる.

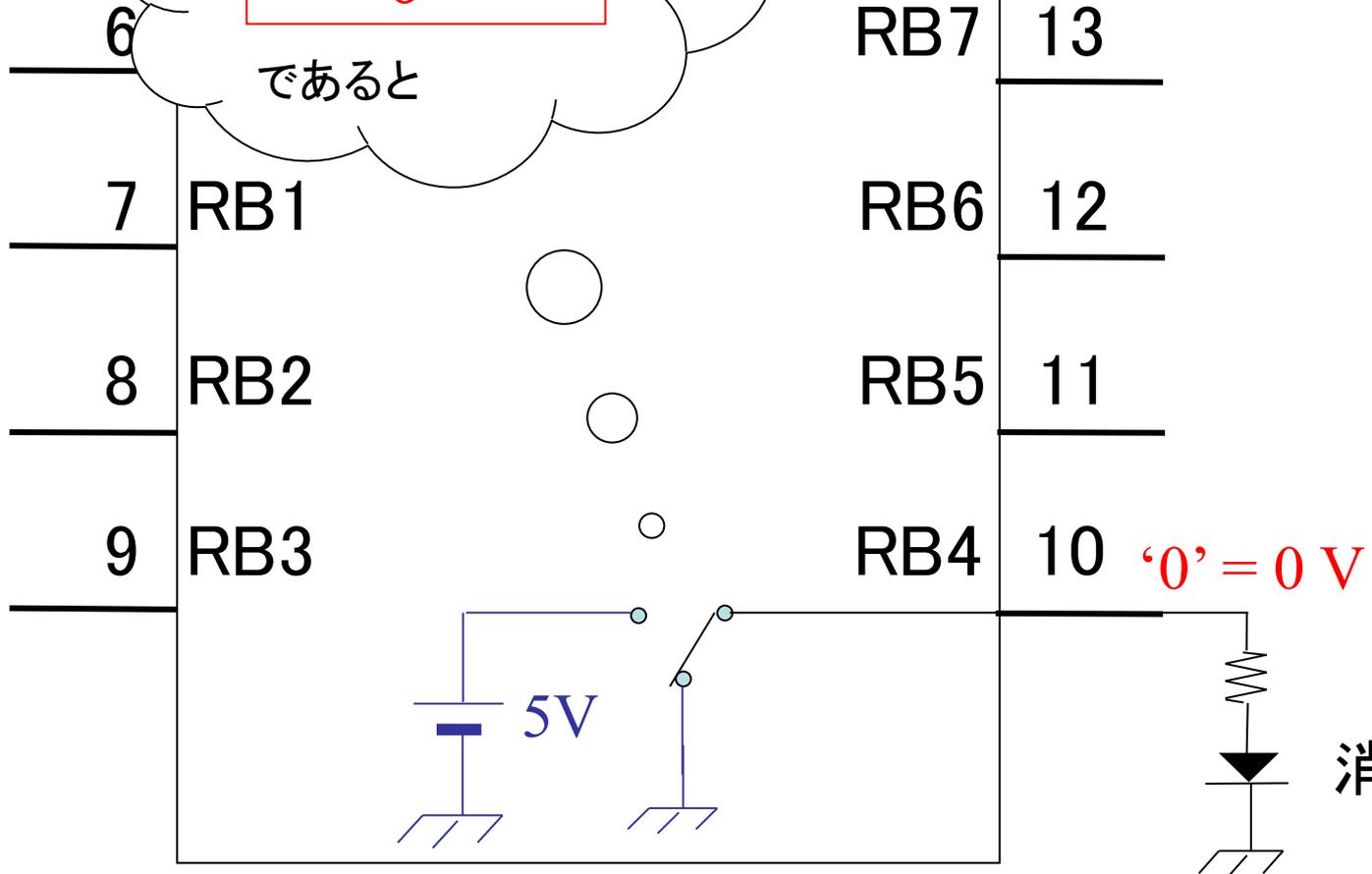


PIC16F84A

PORTBレジスタが

0

であると



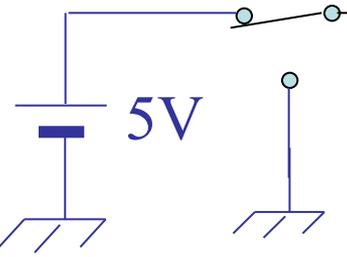
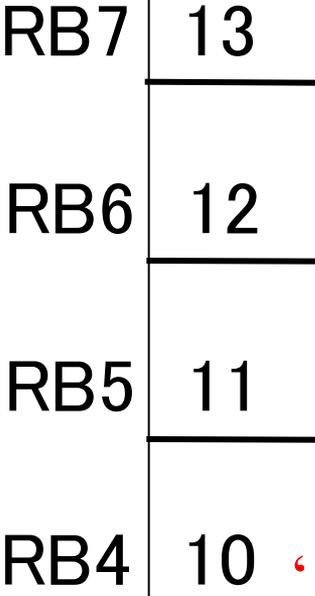
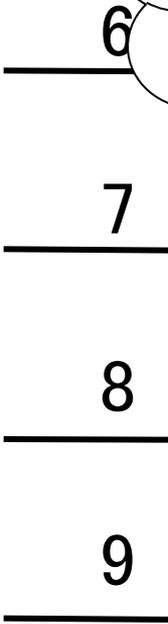
消灯

PIC16F84A

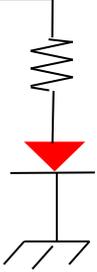
PORTBレジスタが

1

であると



'1' = 5V



; LED control program

```
INCLUDE"p16F84A.inc"  
list p=16F84A
```

```
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
MEM1 EQU 0x0C
```

```
ORG 0
```

```
GOTO START
```

```
ORG 4
```

```
START
```

```
BSF STATUS, RP0
```

```
MOVLW B'00000111'
```

```
MOVWF TRISB
```

```
BCF STATUS, RP0
```

```
STEP1
```

```
MOVF PORTB,0
```

```
ANDLW B'00000111'
```

```
MOVWF MEM1
```

```
RLF MEM1,1
```

```
RLF MEM1,1
```

```
RLF MEM1,1
```

```
RLF MEM1,1
```

```
MOVF MEM1,0
```

```
MOVWF PORTB
```

```
GOTO STEP1
```

```
END
```

BCF f, b Bit Clear f の略です。
ファイル f のビット b を 0 にします。

RP0 を 0 にすると、バンク 0 を選択しま
す。

PORTB レジスタの

RB0-RB2 に入力信号が得られ、

RB3-RB7 から信号を出力できます。

; LED control program

```
INCLUDE "p16F84A.inc"  
list p=16F84A
```

```
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
MEM1 EQU 0x0C
```

```
ORG 0  
GOTO START  
ORG 4
```

```
START
```

```
BSF STATUS,RP0  
MOVLW B'00000111'  
MOVWF TRISB  
BCF STATUS,RP0
```

```
STEP1
```

```
MOVF PORTB,0  
ANDLW B'00000111'  
MOVWF MEM1  
RLF MEM1,1  
RLF MEM1,1  
RLF MEM1,1  
RLF MEM1,1  
MOVF MEM1,0  
MOVWF PORTB  
GOTO STEP1
```

```
END
```

PORTBレジスタの中身をWレジスタへ転送します。

Wレジスタの下位3ビットの値を残し、上位5ビットについては強制的に0とします。

入力情報のみを残すために、このような演算を行います。

例

```
10010110  
AND 00000111  
-----  
00000110
```

; LED control program

```
INCLUDE"p16F84A.inc"  
list p=16F84A
```

```
__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
MEM1 EQU 0x0C
```

```
ORG 0  
GOTO START  
ORG 4
```

```
START
```

```
BSF STATUS,RP0  
MOVLW B'00000111'  
MOVWF TRISB  
BCF STATUS,RP0
```

```
STEP1
```

```
MOVF PORTB,0  
ANDLW B'00000111'
```

```
MOVWF MEM1  
RLF MEM1,1  
RLF MEM1,1  
RLF MEM1,1  
RLF MEM1,1  
MOVF MEM1,0
```

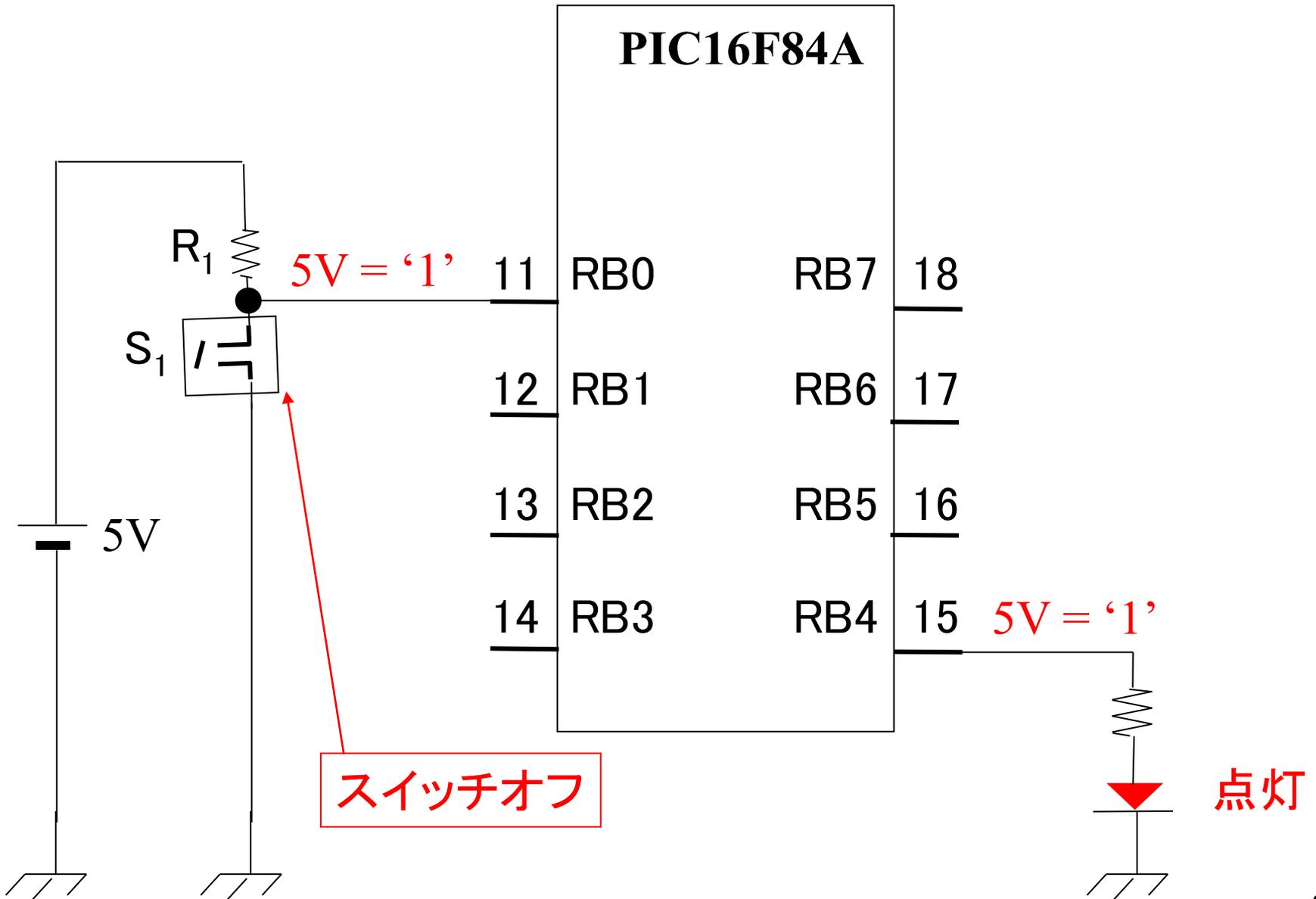
```
MOVWF PORTB  
GOTO STEP1
```

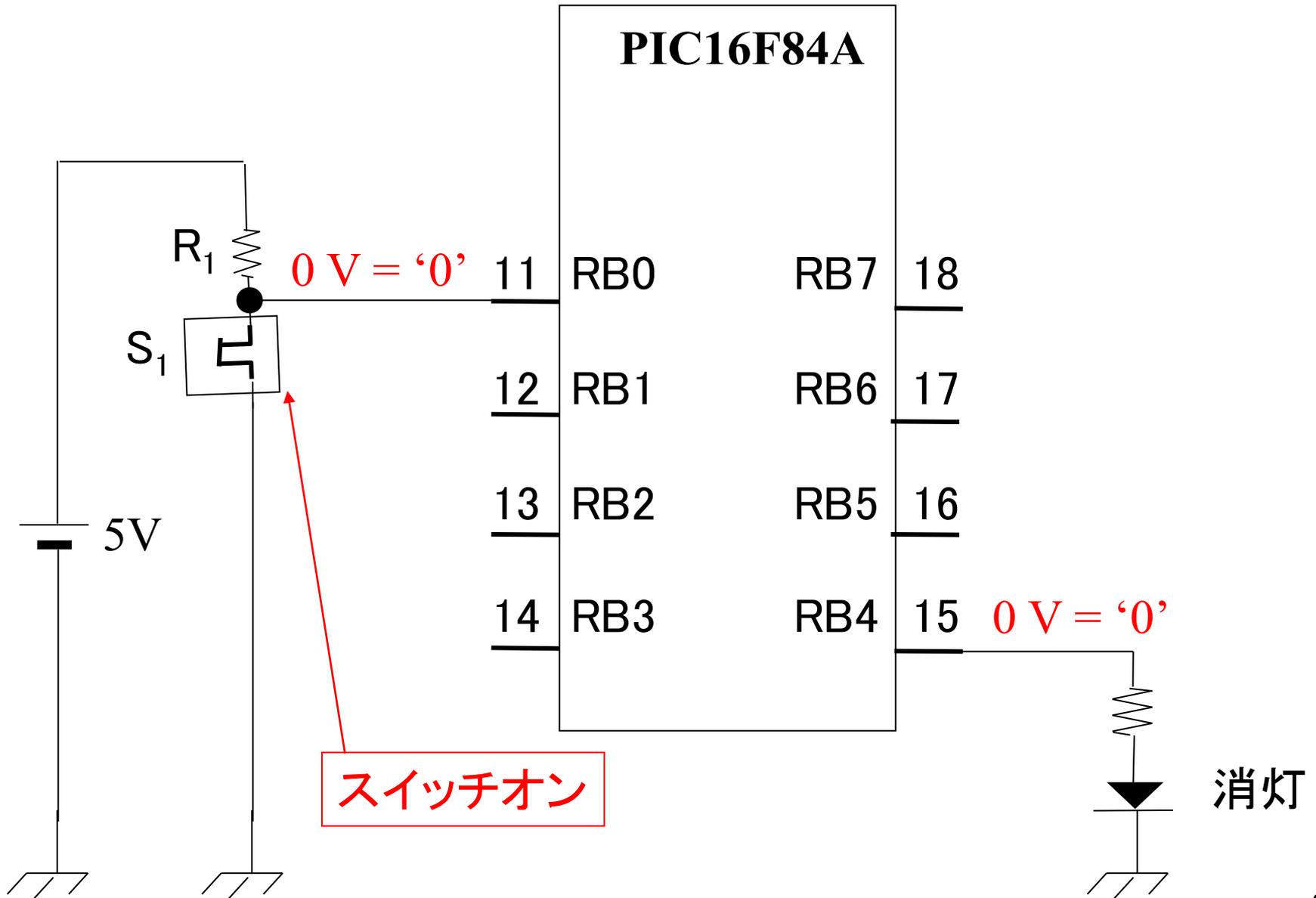
```
END
```

Wレジスタの中身をMEM1に転送して、これを、4ビット回転した後、Wレジスタに戻します。

Wレジスタの中身をPORTBレジスタに転送します。

RB0,1,2の入力は4ビットシフトしてそれぞれRB4,5,6に出力されます。





第0章で製作したマイコンの回路にプログラムの書き込みと実行を行ってください。ブレッドボード上のピンソケットにPICkit3を差し込み、ブレッドボードの電源を入れてください。USBケーブルでパソコンとPICkit3を接続して、Programmer → Select Programmer → PICkit3 により以下のメッセージが出れば、回路は正常に動いています。

The screenshot shows the MPLAB IDE v8.84 interface. The main window displays the assembly code for 'LED_control.asm'. The code includes a header file 'p16F84A.inc' and sets the processor to 'p=16F84A'. It defines memory locations and sets up Port B for output. The program starts by setting Bank 1, configuring TRISB for RB0-2 as input, and then enters a loop (STEP1) that rotates the value in MEM1 and outputs it to PORTB.

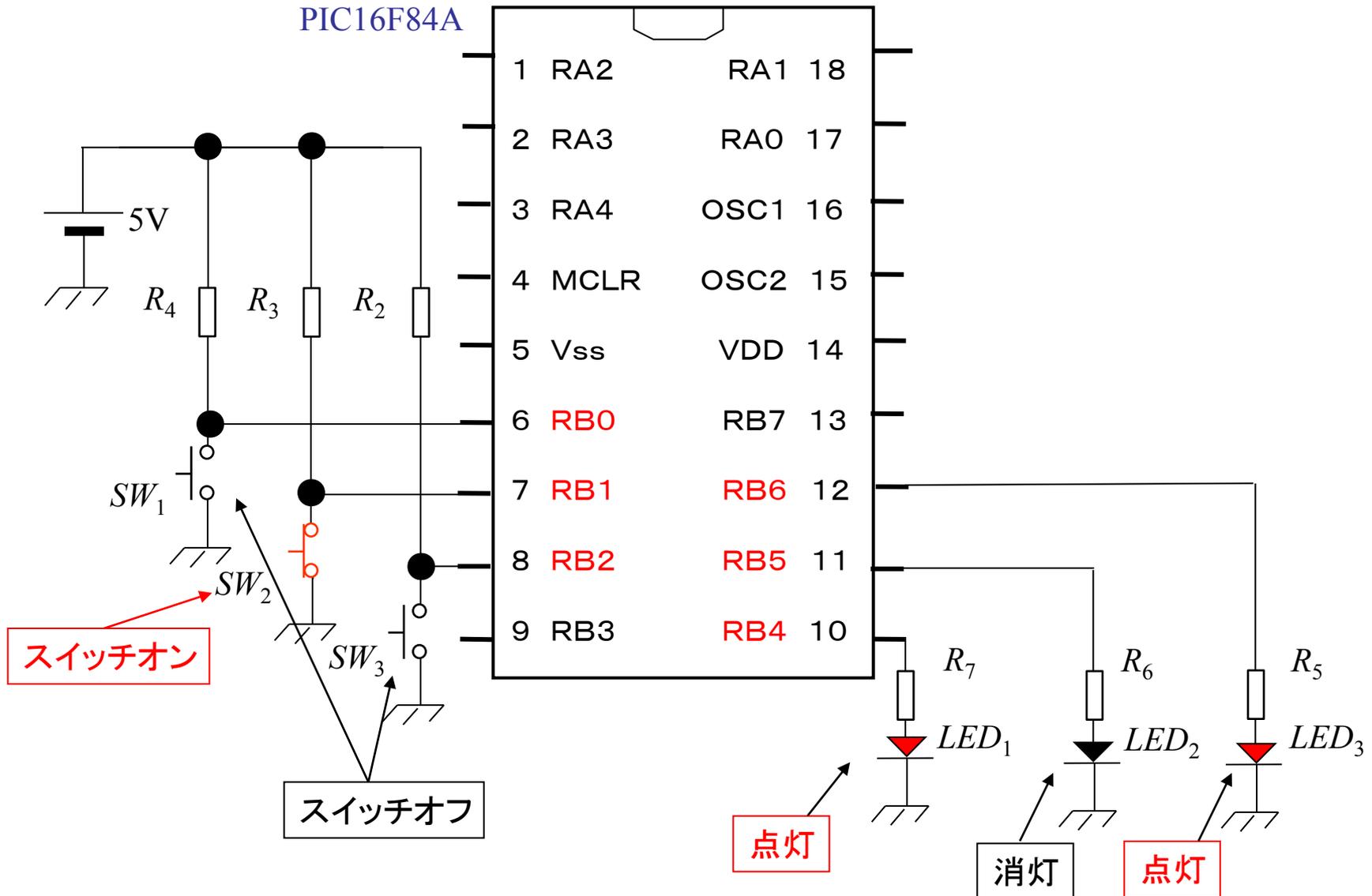
The Output window shows the following messages:

```
PICKit 3 detected
Connecting to PICKit 3...
Firmware Suite Version..... 01.27.04
Firmware type.....Midrange
PICKit 3 Connected.
Target Detected
Device ID Revision = 00000000
```

A blue callout box with a red bracket points to the output messages, containing the text: **これらのメッセージが出ればOK**

The status bar at the bottom indicates: PICKit 3, PIC16F84A, W:0, z dc c, bank 0, Ln 9, Col 1.

Project → Make → Programmer → Program → (ツールバーの右端近くにある)
↑ をクリックするとボード上のマイコンを実行できる. もしくはピンソケットからPICkit 3を引き抜くことでもマイコンを実行できる. 以下のような動作を確認してください.



このソースファイルを打ち込んで下さい。

シミュレータで動作確認が終わったら、PICマイコンにプログラムの書き込みを行い、動作確認をして下さい。

```
; LED control program, modified version  
;
```

```
    INCLUDE "p16F84A.inc"  
    list p=16F84A
```

```
    __CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF
```

```
    ORG      0  
    GOTO    START  
    ORG      4
```

これはスイッチを押すとLEDが点灯するプログラムです。

```
;  
START
```

```
;Setting of Port RB
```

```
    BSF     STATUS,RP0           ;Selection of Bank 1  
    MOVLW  B'0000111'  
    MOVWF  TRISB                ;RB0-2 -> Input Port, RB3-7 -> Output Port  
    BCF     STATUS,RP0           ;Selection of Bank 0
```

```
;  
STEP1  BTFSS  PORTB,0           ;Skip next line, if RB0 = 1  
        GOTO  ST0  
        GOTO  ST1
```

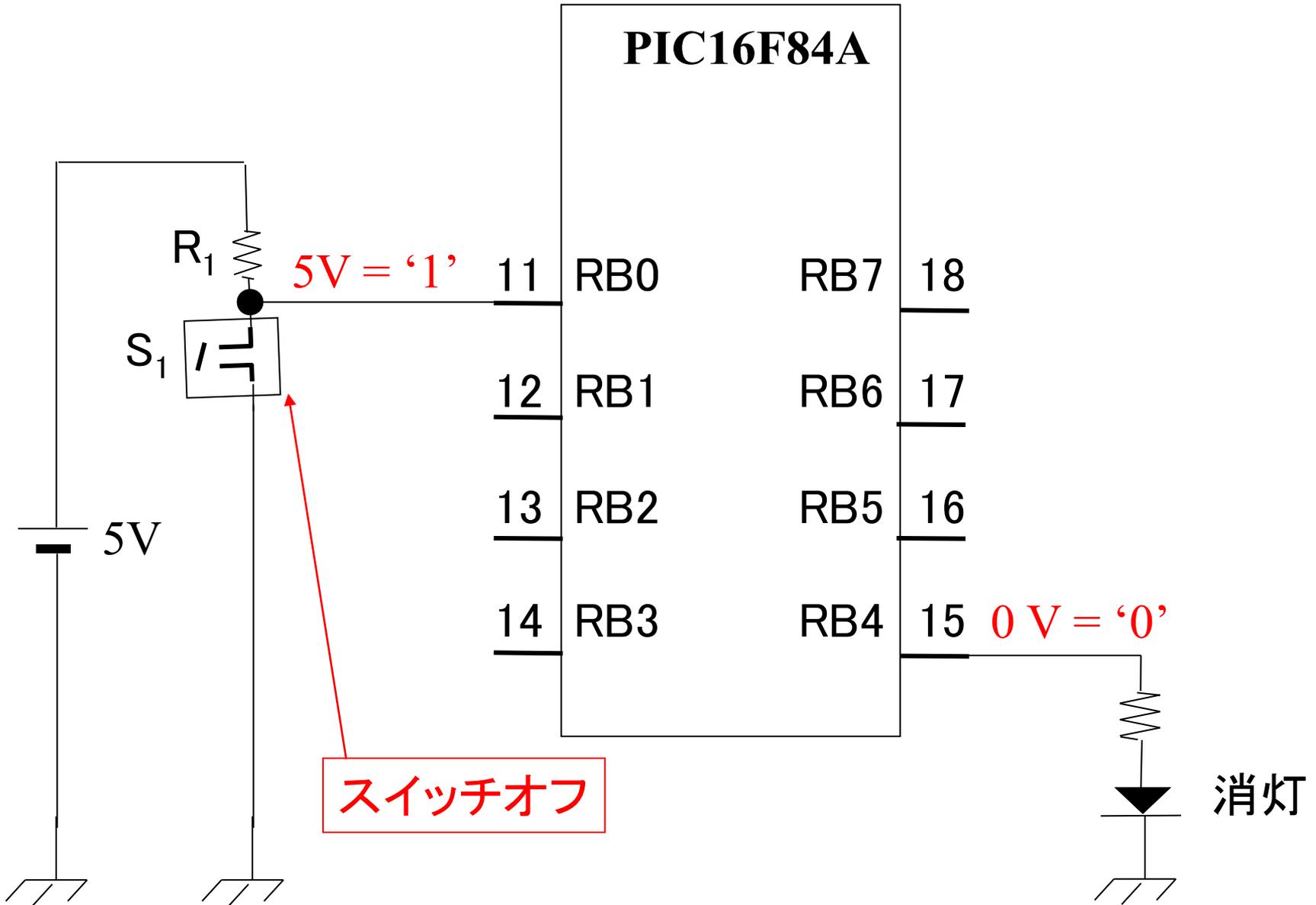
```
ST0     BSF     PORTB,4           ;1 -> RB4
```

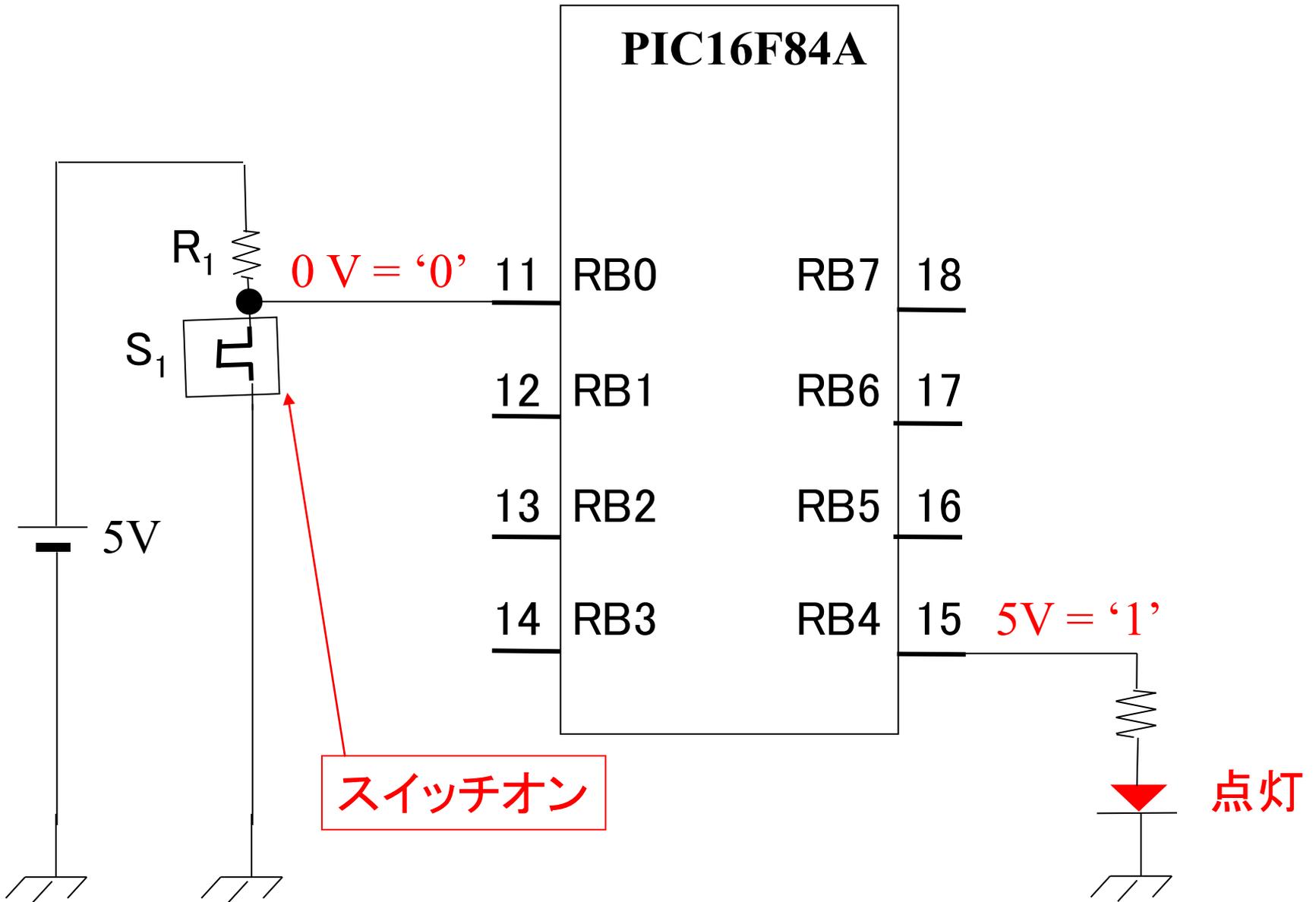
```
        GOTO  STEP2
```

```
ST1     BCF     PORTB,4           ;0 -> RB4
```

```
STEP2   GOTO  STEP1             ;Repetition of Processing
```

```
;  
    END
```





演習問題17 スイッチS1をオンにすると発光ダイオードLED1が点灯し，S1をオフにするとLED1が消灯し，S2をオンにするとLED2が点灯し，S2をオフにするとLED2が消灯し，S3をオンにするとLED3が点灯し，S3をオフにするとLED3が消灯するプログラムを作成せよ。

演習問題18 スイッチS1を押すと発光ダイオードLED1が点灯し，S1をオフにすると約1秒後にLED1が消灯するプログラムを作成せよ。

ヒント：以下のような時間稼ぎサブルーチンを利用する。

```

TIME1 EQU 0x10
TIME2 EQU 0x11
TIME3 EQU 0x12

CALL COUNT1

COUNT1 MOVLW 0x45
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

```

初めにTIME1,2,3に0x45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0x45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0x45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0x45を入れて……とくり返す。TIME1が0になったら終了。0x45 = D'69'より
全部で69×69×69回のくり返し演算を行う。

DECFSZは1サイクル(スキップ時は2サイクル)，GOTOは2サイクル。よって，69回の繰り返しで約69×3サイクルを要する。1サイクルはこのマイコンでは1μsであるので，このサブルーチンは
 $69 \times 69 \times 69 \times 3 \text{ サイクル} \times 1 \mu\text{s} \approx 1 \text{ sec}$

演習問題 1 7 解答例

; LED control program, modified version

;

INCLUDE"p16F84A.inc"

list p=16F84A

__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF

ORG 0

GOTO START

ORG 4

START

;Setting of Port RB

BSF STATUS,RP0 ;Selection of Bank 1

MOVLW B'00000111'

MOVWF TRISB ;RB0-2 -> Input Port, RB3-7 -> Output Port

BCF STATUS,RP0 ;Selection of Bank 0

;

STEP1 BTFSS PORTB,0 ;Skip next line, if RB0 = 1

GOTO STXX0

GOTO STXX1

STXX0 BSF PORTB,4 ;1 -> RB4

GOTO STEP2

STXX1 BCF PORTB,4 ;0 -> RB4

(次ページに続く)

演習問題 1 7 つづき

```
STEP2   BTFSS   PORTB,1           ;Skip next line, if RB1 = 1
         GOTO   STX0X
         GOTO   STX1X

STX0X   BSF    PORTB,5           ;1 -> RB5
         GOTO   STEP3

STX1X   BCF    PORTB,5           ;0 -> RB5

STEP3   BTFSS   PORTB,2           ;Skip next line, if RB2 = 1
         GOTO   ST0XX
         GOTO   ST1XX

ST0XX   BSF    PORTB,6           ;1 -> RB6
         GOTO   STEP4

ST1XX   BCF    PORTB,6           ;0 -> RB6

STEP4   GOTO   STEP1             ;Repetition of Processing
;
        END
```

演習問題 1 8 解答例

; LED control program, modified version

;

INCLUDE"p16F84A.inc"

list p=16F84A

__CONFIG _HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF

TIME1 EQU 0x10

TIME2 EQU 0x11

TIME3 EQU 0x12

ORG 0

GOTO START

ORG 4

START

;Setting of Port RB

BSF STATUS,RP0 ;Selection of Bank 1

MOVLW B'0000111'

MOVWF TRISB ;RB0-2 -> Input Port, RB3-7 -> Output Port

BCF STATUS,RP0 ;Selection of Bank 0

;Main Routine

STEP1 BTFSS PORTB,0 ;Skip next line, if RB0 = 1

CALL ONRTN

BCF PORTB,4 ;0 -> RB4

GOTO STEP1 ;Repetition of Processing

演習問題 1 8 つづき

;Sub Routine 1

```
ONRTN  BSF      PORTB,4      ;1 -> RB4
        BTFSS   PORTB,0      ;Skip next line, if RB0 = 1
        GOTO    ONRTN
        CALL    COUNT1
        RETURN
```

;Sub Routine for Time Consuming for 1 second

```
COUNT1  MOVLW   0x02
        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

2004年8月

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