

## コンピュータ工学 講義プリント(12月18日)

今回は、教科書 P.108 の図 5.2 の LED 実習回路において、2つのスイッチの状態を取得するプログラムの作り方について学習する。

### ・スイッチの状態の取得法(教科書 P.123 参照)

教科書 P.108 の図 5.2 を見れば分かるように、2つのスイッチ SW0 と SW1 がそれぞれ RA0 と RA1 のピンに接続されている。RA0 も RA1 もポート A のピンであるから、TRISA レジスタで RA0 と RA1 を共に入力ピンに設定した後、PORTA を読めば、SW0 と SW1 の状態が取得できる事になる。

ポート A の制御にかかわるレジスタを、次の表 1 に示す。

表 1、ポート A の制御に関するレジスタ

アドレス	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
05H	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0
85H	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0

ポート A の入出力方向を決める TRISA レジスタは、ビット 0 は RA0、ビット 1 は RA1、…という具合に、各ビットに、ポート A のピンが割り当てられている。ポート A のピンは RA0~RA4 までの 5 本なので、TRISA レジスタのビット 5~7 は無効なビットになる。これらの無効なビットに値を書き込んでも無視される。また、これらの無効なビットを読み出すと、必ず 0 が読み出される。

TRISA レジスタの各ビットに 0 を書き込むと、対応するピンが出力ピンになる。また、1 を書き込むと、対応するピンが入力ピンになる。

RA0 と RA1 を入力ピンにしたいのであるから、TRISA の下位 2 ビットは 1 でなければならない。RA2 ~RA4 は空きピンであるから、入力に設定しても出力に設定してもかまわないが、教科書 P.123 の図 5.28 では、これらのピンも入力に設定している。また、TRISA の上位 3 ビットの未使用ビットは、0 を書き込んでも 1 を書き込んでも問題ないが、図 5.28 では 0 を書き込んでいる。

この様に、TRISA レジスタに 00011111B、すなわち 1FH を書き込むと、ポート A の全ピン(RA0~RA4)を入力ピンに設定できる。

TRISA レジスタの設定が終わった後に、PORTA レジスタの下位 2 ビットを読めば、スイッチの状態が分かる。教科書 P.108 の図 5.2 を見れば分かるように、RA0 と RA1 の信号線は、それぞれ 10kΩ の抵抗でプルダウンされており、また、+5V と RA0、RA1 の信号線の間、それぞれ SW0、SW1 の 2 つのスイッチが接続されている。

すなわち、スイッチが OFF の時は、RA0 や RA1 にはプルダウン抵抗により 0 が入力され、スイッチが ON の時は、RA0 や RA1 はスイッチにより +5V に接続されるため、1 が入力される。

よって、PORTA のビット 0 が 0 なら SW0 は OFF、1 なら SW0 は ON と判断できる。同様に PORTA のビット 1 が 0 なら SW1 は OFF、1 なら SW1 は ON と判断できる。

### ・スイッチ入力による LED の制御(教科書 P.123 参照)

教科書 P.123 の表 5.1 の様に、SW0 と SW1 により、LED の点灯状態を制御するプログラムを作る事にする。この様なプログラムを作れば、LED の点灯状態を確認する事により、PIC16F84A が、正しくスイッ

チの状態を読み取れているかどうか判断できる。

LEDの点灯パターンを決めるには、教科書 P.124 の図 5.30 のフローチャートに示すように、まず SW0 の状態(RA0)を読み取り、LEDの点灯信号の下位4ビットを決定する。すなわち、SW0がOFFなら、この時点での点灯信号は00Hであり、SW1がONなら、この時点での点灯信号は0FHである。

次に SW1 の状態(RA1)を読み取り、SW1がONなら、点灯信号と F0H の論理和を取って、上位4ビットを全て1にする。

この様な考え方で作ったのが、教科書 P.124 のリスト 5.7 の、スイッチ入力プログラムである。このプログラムは、電源投入時に一度だけスイッチの状態を読み取り、それを LED の点灯パターンで表示する。すなわち、プログラムを実行した後に SW0 や SW1 の状態を変化させても LED の点灯状態には反映されない。

これではデモに不都合であるので、SW0 や SW1 の状態を、すぐに LED の点灯状態に反映するようにしたプログラムが、次のリスト 1 である。

リスト 1、リスト 5.7 を書き換え、スイッチの状態がすぐに LED に反映する用にしたプログラム

```
LIST    P=PIC16F84A
INCLUDE "P16F84A.INC"

ORG     0

BSF     STATUS, RPO    ; バンク 1 を選択
MOVLW   1FH           ; 0001111B を W レジスタに代入
MOVWF   TRISA         ; ポート A を全て入力モードに設定
CLRF    TRISB         ; ポート B を全て出力モードに設定
BCF     STATUS, RPO    ; バンク 0 を選択
CLRF    PORTB         ; ポート B を 0 にクリア (LED 消灯)

LOOP    CLRW           ; W レジスタをクリア
        BTFSC PORTA, 0 ; RA0 をチェック
        MOVLW 0FH      ; 1 (ON) ならば、W←00001111B
        BTFSC PORTA, 1 ; RA1 をチェック
        IORLW 0F0H     ; 1 (ON) ならば、W←W OR 11110000B
        MOVWF PORTB    ; 点灯データをポート B に出力
        GOTO  LOOP     ; LOOP にジャンプ

END
```

リスト 1 を、教科書のリスト 5.7 と比較すると、違うのは 2 行だけである。

まず、教科書では単に CLRW となっていた行が、LOOP CLRW と、ラベルがつくようになった。

次に、教科書では WAIT GOTO WAIT となっていた行が GOTO LOOP となり、スイッチの読み取りと LED の表示を無限ループで繰り返すようになった。

この 2 行の変更により、スイッチの状態を変化させると、すぐに LED の点灯状態に反映されるようになった。

講義では、このプリントのリスト 1 のプログラムのデモを行うので、スイッチを切り替えると、LED の点灯状態が変化の様子を各自で確認する事。

- ・光が流れるプログラム(片道バージョン)の光が流れる速さを、スイッチで変える

前回の講義で、光が流れるプログラム(片道バージョン)というのを作った。点灯する LED が左から右に順に移動していき、一番右の LED が光り終わると、次は一番左の LED に戻るプログラムであった。このプログラムでは、光が移動する時間間隔は 0.5 秒であったが、今回は、2 つのスイッチ SW0 と SW1 の操作により、光が移動する時間間隔を変える事ができる様に、プログラムを組みなおしてみよう。

光が移動する時間間隔は、次の表 2 の様になるとする。

表 2、スイッチの状態と、光が移動する時間間隔

SW1(RA1)の状態	SW0(RA0)の状態	時間間隔(秒)
OFF(0)	OFF(0)	0.01
OFF(0)	ON(1)	0.03
ON(1)	OFF(0)	0.10
ON(1)	ON(1)	0.30

前回の講義で説明した、光が流れるプログラム(片道バージョン)のプログラムを、次のリスト 2 に示す。

リスト 2、光が流れるプログラム(片道バージョン、時間間隔固定)

```
LIST    P=PIC16F84A
INCLUDE "P16F84A.INC"
```

```
LEDD    EQU    80H        ; LED 点灯データの設定
CNT1    EQU    0CH        ; タイマ 1 用カウンタ変数
CNT2    EQU    0DH        ; タイマ 2 用カウンタ変数
CNT3    EQU    0EH        ; タイマ 3 用カウンタ変数

ORG     0

BSF     STATUS, RPO      ; バンク 1 を選択
CLRF   TRISB            ; ポート B を全て出力モードに設定
BCF     STATUS, RPO      ; バンク 0 を選択
BCF     STATUS, C        ; C フラグをクリア

MOVLW  LEDD             ; 点灯データを W レジスタにセット
MOVWF  PORTB            ; 点灯データをポート B に出力
```

```

REPEAT CALL  TIMER3      ; 0.5 秒タイマの呼び出し
        RRF    PORTB, 1   ; ポート B を 1 ビット右にローテイト
        BTFSZ  STATUS, C  ; C フラグが 0 ならば、次の命令をスキップ
        RRF    PORTB, 1   ; C フラグが 1 ならば、もう一度、右にローテイト
        GOTO   REPEAT

TIMER1  MOVLW  D' 62'     ; 0.1 ミリ秒タイマサブルーチン
        MOVWF  CNT1

LOOP1   NOP
        DECFSZ CNT1, 1
        GOTO   LOOP1
        RETURN

TIMER2  MOVLW  D' 100'    ; 10 ミリ秒タイマサブルーチン
        MOVWF  CNT2

LOOP2   NOP
        CALL   TIMER1
        DECFSZ CNT2, 1
        GOTO   LOOP2
        RETURN

TIMER3  MOVLW  D' 50'     ; 0.5 秒タイマサブルーチン
        MOVWF  CNT3

LOOP3   NOP
        CALL   TIMER2
        DECFSZ CNT3, 1
        GOTO   LOOP3
        RETURN

        END

```

TIMER3 は、10ms のタイマ(TIMER2)を 50 回呼び出すことで、500ms(0.5 秒)のタイマを実現している。これを、ウェイト時間が可変のタイマに書き換えたい。

TIMER3 の最初の 2 命令(MOVLW D'50' と MOVWF CNT3)で変数 CNT3 を 50 に初期化することにより、TIMER2 の呼び出し回数を 50 回に設定しているが、この初期化をなくして、メインルーチンの方で CNT3 を初期化してから TIMER3 を呼び出すようにすると、CNT3 の初期値×10ms のタイマサブルーチンができる。また、NOP 命令は特に意味がないので、削除してよい。この様にして作ったタイマサブルーチンが次のリスト 3 である。

リスト 3、ウェイト時間が可変のタイマルーチン

```
TIMER3    CALL    TIMER2    ; CNT3×10 ミリ秒タイマサブルーチン
          DECFSZ CNT3,1
          GOTO    TIMER3
          RETURN
```

次に考えることは、SW0 と SW1 の状態によって、CNT3 の初期値を変える事である。

MOVFPOR TA,0 を実行すると、W レジスタにポート A の状態が読み込まれる。ポート A の下位 2 ビットである RA1、RA0 が、スイッチの状態を表わしている。RA2～RA4 までは、何も接続されていない入力端子の状態を表わしているの、0 か 1 かは不定である。(静電気の分布などにより決まる)

よって、W レジスタの値と 03H の論理積を取り、上位 6 ビットを強制的に 0 にすると、W レジスタの値はスイッチの状態のみで決まる事になる。これをアセンブリ言語で書くと ANDLW 03H となる。

この時もし W=0 なら、RA1=0 かつ RA0=0 であるから、CNT3 を 1 に初期化すればよい。

そうでなければ、W から 1 を引いてみて(ADDLW OFFH)、W=0 なら、RA1=0 かつ RA0=1 であるから、CNT3 を 3 に初期化すればよい。

そうでなければ、さらに W から 1 を引いてみて、W=0 なら、RA1=1 かつ RA0=0 であるから、CNT3 を 10 に初期化すればよい。

そうでなければ、RA1=1 かつ RA0=1 であるから、CNT3 を 30 に初期化すればよい。

この様な考え方で、スイッチの状態に応じたウェイト時間を実現するルーチンが次のリスト 4 である。

#### リスト 4、スイッチの状態に応じてウェイト時間を変化させるルーチン

```
REPEAT    MOVF    PORTA, 0    ; PORTA→W
          ANDLW   03H        ; W AND 03H→W (下位 2 ビット以外をマスク)
          BTFSS  STATUS, Z    ; W=0 ならスキップ
          GOTO   CHECK1      ; CHECK1 へジャンプ
CASE0     MOVLW  D' 1'       ; 1→W
          GOTO   SET_TIMER   ; SET_TIMER へジャンプ
CHECK1    ADDLW  OFFH        ; W-1→W
          BTFSS  STATUS, Z    ; W=0 ならスキップ
          GOTO   CHECK2      ; CHECK2 へジャンプ
CASE1     MOVLW  D' 3'       ; 3→W
          GOTO   SET_TIMER   ; SET_TIMER へジャンプ
CHECK2    ADDLW  OFFH        ; W-1→W
          BTFSS  STATUS, Z    ; W=0 ならスキップ
          GOTO   CASE3       ; CASE3 へジャンプ
CASE2     MOVLW  D' 10'      ; 10→W
          GOTO   SET_TIMER   ; SET_TIMER へジャンプ
CASE3     MOVLW  D' 30'      ; 30→W
SET_TIMER MOVWF  CNT3        ; W→CNT3
          CALL   TIMER3      ; タイマの呼び出し
          BCF   STATUS, C     ; C フラグをクリア
```

リスト 4 のルーチンでは、ADDLW 命令を使う事で、C フラグを書き換えてしまう。C=1 になると、後続の RRF 命令に悪影響するので、最後に C フラグをクリアしている事に注意されたい。また、ここで C フラグをクリアすると、REPEAT ラベルで始まるメインループの前には、C フラグのクリアの命令は必要なくなる。

以上の事を考慮して、リスト 2 を、スイッチの状態に応じて光の流れる速さを変えられるように作り変えると、次のリスト 5 の様になる。

リスト 5、光が流れるプログラム(片道バージョン、時間間隔可変)

```

LIST    P=PIC16F84A
INCLUDE "P16F84A.INC"
__CONFIG _FOSC_HS & _WDTE_OFF & _PWRTE_ON & _CP_OFF

LEDD    EQU    80H        ; LED 点灯データの設定
CNT1    EQU    0CH        ; タイマ 1 用カウント変数
CNT2    EQU    0DH        ; タイマ 2 用カウント変数
CNT3    EQU    0EH        ; タイマ 3 用カウント変数

ORG     0

BSF     STATUS, RPO      ; バンク 1 を選択
MOVLW   1FH              ; 1FH→W
MOVWF   TRISA            ; ポート A を全て入力モードに設定
CLRF    TRISB            ; ポート B を全て出力モードに設定
BCF     STATUS, RPO      ; バンク 0 を選択

MOVLW   LEDD             ; 点灯データを W レジスタにセット
MOVWF   PORTB            ; 点灯データをポート B に出力
REPEAT  MOVF   PORTA, 0   ; PORTA→W
        ANDLW  03H        ; W AND 03H→W (下位 2 ビット以外をマスク)
        BTFSS STATUS, Z   ; W=0 ならスキップ
        GOTO   CHECK1     ; CHECK1 ヘジャンプ
CASE0   MOVLW  D' 1'      ; 1→W
        GOTO   SET_TIMER  ; SET_TIMER ヘジャンプ
CHECK1  ADDLW  OFFH        ; W-1→W
        BTFSS STATUS, Z   ; W=0 ならスキップ
        GOTO   CHECK2     ; CHECK2 ヘジャンプ
CASE1   MOVLW  D' 3'      ; 3→W
        GOTO   SET_TIMER  ; SET_TIMER ヘジャンプ
CHECK2  ADDLW  OFFH        ; W-1→W

```

```

        BTFSS STATUS, Z      ; W=0 ならスキップ
        GOTO CASE3          ; CASE3 へジャンプ
CASE2   MOVLW D' 10'        ; 10→W
        GOTO SET_TIMER     ; SET_TIMER へジャンプ
CASE3   MOVLW D' 30'        ; 30→W
SET_TIMER MOVWF CNT3        ; W→CNT3
        CALL TIMER3        ; タイマの呼び出し
        BCF STATUS, C      ; C フラグをクリア
        RRF PORTB, 1       ; ポート B を 1 ビット右にローテイト
        BTFSC STATUS, C    ; C フラグが 0 ならば、次の命令をスキップ
        RRF PORTB, 1       ; C フラグが 1 ならば、もう一度、右にローテイト
        GOTO REPEAT

TIMER1  MOVLW D' 62'        ; 0.1 ミリ秒タイマサブルーチン
        MOVWF CNT1
LOOP1   NOP
        DECFSZ CNT1, 1
        GOTO LOOP1
        RETURN

TIMER2  MOVLW D' 100'       ; 10 ミリ秒タイマサブルーチン
        MOVWF CNT2
LOOP2   NOP
        CALL TIMER1
        DECFSZ CNT2, 1
        GOTO LOOP2
        RETURN

TIMER3  CALL TIMER2        ; CNT3×10 ミリ秒タイマサブルーチン
        DECFSZ CNT3, 1
        GOTO TIMER3
        RETURN

        END

```

講義では、リスト 5 のプログラムのデモを行うので、光が流れる速さが、スイッチの操作で変わる様子を、各自確認する事。