Report on the Experiment

No. 2

Subject ワンチップマイコン基礎実験II

Date 2020. 12. 23

Weather 晴れ Temp 24 ℃ Wet 65.5 %

Class E4
Group
Chief
Partner

No 14
Name 小畠 一泰

Kure National College of Technology

1　目的

PIC16F84A を例に「プログラムカウンタの利用」と「割り込み処理」について理解し, 7 セグメント LED の使用方法に習熟することを, また後半では, サウンダを駆動することを目的とする.

2　課題

下記ソースコードを共通部分とし, すべてのソースコードは下記を含むものとする.

コード 1 共通部分

```
  list p=16F84A
  #include <p16F84A.inc>
  __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _HS_OSC

  ORG 0x000

  bsf STATUS, RP0 ; バンク 0 → 1 に切り替え
  clrf TRISB      ;TRISB をすべて 0, 全ビット出力設定に
  bcf STATUS, RP0 ; バンク 1 → 0 に切り替え

  clrf PORTB      ; すべて消灯
```

また下記を標準タイマーとし, 明示しない限りこれらも含み使用できるものとする.

コード 2 タイマー

```
  COUNTER1 EQU 0x0C  ; 作業用変数の設定
  COUNTER2 EQU 0x0D  ; 汎用ファイルレジスタ 0Ch から

WAIT
  clrf COUNTER1 ; 変数 1 をクリア

WAIT1
  clrf COUNTER2      ; 変数 2 をクリア
  call WAIT2         ; さらにサブルーチンへ
  decfsz COUNTER1, 1 ; 変数から 1 を引く
  goto WAIT1         ; 結果が 0 でなければ実行
  return             ; 結果が 0 ならばスキップしこの行を実行

WAIT2
  decfsz COUNTER2, 1 ; サブルーチンのサブルーチン
  goto WAIT2
  return
```

コード **3** [課題 1] 表 1 を完成させ, List6 の欠落している部分を補いプログラムを完成させよ

```
1  movlw B'00011111'    ; 下位 5 ビットを 1 に
2  movwf TRISA          ;TRISA に代入, RA0 から RA4 が入力設定に
3  movlw B'00000000'    ;W レジスタにすべて 0 に
4  movwf TRISB          ;TRISB をすべて 0, 出力設定に
5  bcf STATUS, RP0      ; バンク 1 → 0 に切り替え
6
7  clrw                 ;W レジスタをクリア
8
9  MAIN
10   bcf STATUS,C       ;C フラグクリア
11   movf PORTA, 0      ; 入力データを W レジスタに転送
12   call SS_Data       ;7 セグ LED 点灯データ取得ルーチンに
13   movwf PORTB        ; データ表示
14   goto MAIN
15
16 ;7 セグ LED 点灯データ取得ルーチン
17 SS_Data
18   addwf PCL,1        ;W+PCL(プログラムカウンタの下位) → PCL
19   retlw B'00111111' ;W =  0 →左の点灯データを W レジスタが持って帰る
20   retlw B'00000110' ;W =  1 →左の点灯データを W レジスタが持って帰る
21   retlw B'01011011' ;W =  2
22   retlw B'01001111' ;W =  3
23   retlw B'01100110' ;W =  4
24   retlw B'01101101' ;W =  5
25   retlw B'01111101' ;W =  6
26   retlw B'00000111' ;W =  7
27   retlw B'01111111' ;W =  8
28   retlw B'01101111' ;W =  9
29   retlw B'01110111' ;W = 10 → "A"のデータを持って帰る
30   retlw B'01111100' ;W = 11 → "b"のデータを持って帰る
31   retlw B'01011000' ;W = 12 → "c"のデータを持って帰る
32   retlw B'01011110' ;W = 13 → "d"のデータを持って帰る
33   retlw B'01111001' ;W = 14 → "E"のデータを持って帰る
34   retlw B'01110001' ;W = 15 → "F"のデータを持って帰る
35
36   END
```

コード 4 [課題 2] 各自が自由に割り込み処理を確認するプログラムを作成せよ

```
;割り込みサブルーチンのみ抜粋（他は List 7と同じ）
SUB
  nop

  movlw 0x01
  andwf PORTB,1 ;PORTB の RB0 以外をクリア
  call WAIT
  movlw 0x08    ;RB3 のみ 1
  addwf PORTB,1 ;W を加算して PORTB に上書き，RB4 のみ 1
  call WAIT

  movlw 0x01
  andwf PORTB,1
  call WAIT
  movlw 0x04    ;RB2 のみ 1
  addwf PORTB,1
  call WAIT;

  nop
  return
```

コード 5 [課題 3] 1kHz のパルス (0.5ms を ON, 0.5ms を OFF) を RA3 に出力するプログラム

```
  clrf TRISA  ;TRISA をすべて 0，全ビット出力設定に
  clrf PORTA

MAIN
  bsf PORTA, 3        ;RA3 を ON
  call WAIT           ;0.5ms 待機

  bcf PORTA, 3        ;RA3 を OFF
  call WAIT           ;0.5ms 待機

  goto MAIN           ;無限ループ

WAIT
  movlw D'249'
  movwf COUNTER1      ;変数 1 をクリア

WAIT1
  nop
  nop
  decfsz COUNTER1, 1 ;変数から 1 を引く
  goto WAIT1          ;結果が 0 でなければ実行
  return              ;結果が 0 ならばスキップしこの行を実行
```

コード **6** [課題 4] メロディを作成せよ (その 1)

```
  COUNTER1 EQU 0x0C;
  COUNTER2 EQU 0x0D;
  COUNTER3 EQU 0x0E;
  COUNTER4 EQU 0x0F;
  COUNTER5 EQU 0x10;
  COUNTER6 EQU 0x11;
  COUNTER7 EQU 0x12;
  COUNTER8 EQU 0x13;
  COUNTER9 EQU 0x14;

  ORG 0x000

  bsf STATUS, RP0
  clrf TRISA
  bcf STATUS, RP0
  clrf PORTA

MAIN
  call PLAY_C
  call PLAY_D
  call PLAY_E
  goto MAIN

;==== IF ====
PLAY_C
  clrf COUNTER7
  call PLAY_C_

PLAY_D
  clrf COUNTER8
  call PLAY_D_

PLAY_E
  clrf COUNTER9
  call PLAY_E_
```

コード 7 [課題 4] メロディを作成せよ (その 2)

```
;==== private methods ====
PLAY_C_
  bsf PORTA, 3
  call WAIT1
  bcf PORTA, 3
  call WAIT1
  decfsz COUNTER7, 1
  goto PLAY_C_
  return

PLAY_D_
  bsf PORTA, 3
  call WAIT4
  bcf PORTA, 3
  call WAIT4
  decfsz COUNTER8, 1
  goto PLAY_D_
  return

PLAY_E_
  bsf PORTA, 3
  call WAIT7
  bcf PORTA, 3
  call WAIT7
  decfsz COUNTER9, 1
  goto PLAY_E_
  return

;==== timer ====
;==== timer - C ====
WAIT1
  movlw D'176'
  movwf COUNTER1

WAIT2
  movlw D'02'
  movwf COUNTER2
  call WAIT3
  decfsz COUNTER1, 1
  goto WAIT2
  return

WAIT3
  decfsz COUNTER2, 1
  goto WAIT3
  return
```

コード 8 [課題 4] メロディを作成せよ (その 3)

```
;==== timer - D ====

WAIT4
  movlw D'160'
  movwf COUNTER3

WAIT5
  movlw D'02'
  movwf COUNTER4
  call WAIT6
  decfsz COUNTER3, 1
  goto WAIT5
  return

WAIT6
  decfsz COUNTER4, 1
  goto WAIT6
  return

;==== timer - E ====

WAIT7
  movlw D'144'
  movwf COUNTER5

WAIT8
  movlw D'02'
  movwf COUNTER6
  call WAIT9
  decfsz COUNTER5, 1
  goto WAIT8
  return

WAIT9
  decfsz COUNTER6, 1
  goto WAIT9
  return
```

3 調査課題

1. PIC16F84A の EEPROM(Electrically Erasable and Programmable Read Only Memory) はレジスタファイル空間に直接マッピングされていないので, 特殊機能レジスタを通じて間接的にアドレスしてリード/ライトします. この使い方 (書き込み, 読み出し) の手順について調べ, プログラム例を示して説明せよ.
   - 書き込み
     1. EEADR レジスタに書き込むべきアドレスを指定する
     2. EEDATA レジスタに書き込むデータを設定する
     3. EECON1 の 2 ビット目 (WREN) を 1 にする (書き込み有効にする)
     4. EECON2 に 55H, AAH を書き込む
     5. EECON1 の 1 ビット目 (WR) を 1 にする (書き込み要求を行う)
     6. 書き込みが終わるのを待つ

コード **9** 書き込みのプログラム例

```
1 BSF STATUS, RP0   ; バンク 1 に切り替える
2 BCF INTCON, GIE   ; 割り込み禁止
3 BSF EECON1, WREN  ; 書き込み有効
4 MOVLW 55h
5 MOVWF EECON2      ;55h(番地) の書き込み
6 MOVLW AAh
7 MOVWF EECON2      ;AAh(値) の書き込み
8 BSF EECON1, WR    ; 書き込み要求を立てる
9 BSF INTCON, GIE   ; 割り込み許可
```

   - 読み出し
     1. EEADR に読み出したい場所のアドレスをセットする
     2. EECON1 の 0 ビット目 (RD ビット) を 1 にする
     3. EEDATA からデータを読み取る

コード **10** 読み出しのプログラム例

```
1 BCF STATUS, RP0   ; バンク 0 に切り替える
2 MOVLW 0x00        ; 読み込む EEPROM のアドレスを指定する (この例では 0 番地)
3 MOVWF EEADR       ; 指定したアドレスの読み込み
4 BSF STATUS, RP0   ; バンク 1 に切り替える
5 BSF EECON1, RD    ; 読み出し要求を立てる
6 BCF STATUS, RP0   ; バンク 0 に切り替える
7 MOVF EEDATA, W    ;W レジスタに EEPROM の内容を読み出す
```

2. 7 セグメント LED を 2 桁表示させる方法について, 回路図を”bsch” などを用いて描き, 上位桁に 1, 下位桁に 2 を表示するプログラムを作成せよ.

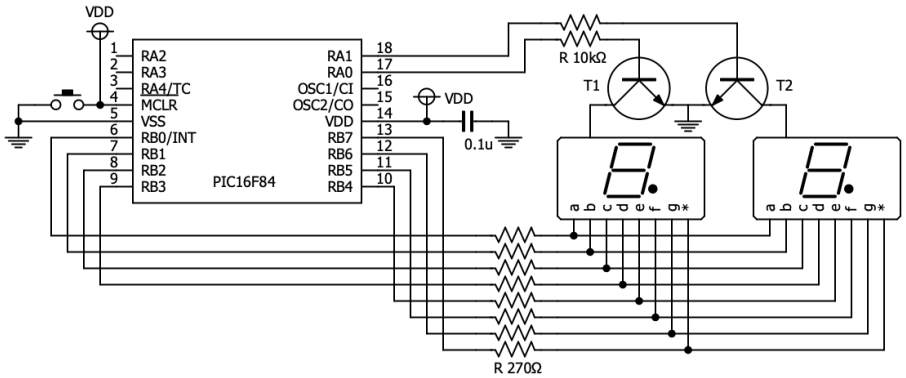

図 1: 回路図

コード **11** 7 セグメント LED で 2 桁表示 (その 1)

```
  list p=16F84A
  #include <P16F84A.INC>
  __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _HS_OSC
  ORG 0x000

  bsf STATUS, RP0       ; バンク 0 → 1 に切り替え
  movlw B'00000000'     ;W レジスタに 0 を代入
  movwf TRISA           ;PORTA を出力設定に
  movlw B'00000000'     ;W レジスタに 0 を代入
  movwf TRISB           ;PORTB を出力設定に
  bcf STATUS, RP0       ; バンク 1 → 0 に切り替え
  clrw                  ;W レジスタをクリア
  clrf PORTA
  clrf PORTB

MAIN
  movlw B'00000110'  ;W = 1
  call USE_DIG1_7SEG
  call   WAIT

  movlw B'01011011'  ;W = 2
  call USE_DIG2_7SEG
  call   WAIT

  goto MAIN
```

コード **12** 7 セグメント LED で 2 桁表示 (その 2)

```
;右側の 7セグを非表示→ Wレジスタから Fレジスタへ移動→左側の 7セグを表示
USE_DIG1_7SEG
  bcf PORTA, 1
  movwf PORTB
  bsf PORTA, 0
  return

;左側の 7セグを非表示→ Wレジスタから Fレジスタへ移動→右側の 7セグを表示
USE_DIG2_7SEG
  bcf PORTA, 0
  movwf PORTB
  bsf PORTA, 1
  return

;50ms timer
WAIT
  movlw D'100'
  movwf COUNTER1

;0.5ms timer
WAIT1
  movlw D'249'
  movwf COUNTER2
  decfsz COUNTER2, 1 ;変数から 1を引く
  goto WAIT2         ;結果が 0でなければ実行
  return             ;結果が 0ならばスキップしこの行を実行

WAIT2
  nop
  nop
  decfsz COUNTER2, 1 ;変数から 1を引く
  goto WAIT2         ;結果が 0でなければ実行
  return             ;結果が 0ならばスキップしこの行を実行
```

[解説] 人間の目の残像効果を利用し, 本プログラムでは 50ms で 7 セグの入れ替えを行うダイナミック点灯方式を採用したことで, 常時 2 つの 7 セグが点灯しているかのように見せることにした. また 7 セグはカソードコモンタイプを利用し, RA0,RA1 のビットを立てることでトランジスタのスイッチを操作し, 表示するべき 7 セグが表示されるようにしている.

3. 割り込み条件はいくつもあるが, PIC16F84A に許可されている割り込み条件のうち,「ポート RB0 からの外部割込み」「TMR0 カウンタオーバフロー内部割込み」, それぞれに対する特殊レジスタの設定について説明せよ.

表 1: INTCON レジスタの詳細

ラベル	役割
INTF	RB0/INT の割込みフラグ
INTE	RB0/INT の割込み許可ビット
T0IF	TMR0 タイマ割込みフラグ
T0IE	TMR0 タイマの割込み許可ビット
GIE	全体割込み許可ビット

- ポート RB0 からの外部割込み
  RB0 ポートの入力信号の立ち上がり（または立ち下がり）のエッジで割込みを発生. 立ち上がり／立ち下がりエッジの指定には OPTION レジスタの INTEDG ビットで行う. INTEDG = 0 の際は立ち下がりエッジで割込み, INTEDG = 1 の際は立ち上がりエッジで割込む. 割込みの許可には INTCON レジスタの INTE ビットを 1 にして割込みを許可し, その後 GIE ビットを 1 にすれば割込み待ちとなる. また割込んだらフラグのクリアは割込み処理へジャンプしてきた際に, INTCON レジスタの INTF フラグをクリアして次の割込みに備える.
- TMR0 カウンタオーバフロー内部割込み
  カウンタへの条件設定などした時には常に TMR0 はゼロクリアされる. カウンタがオーバーフローした時にはオーバーフローフラグとして INTCON レジスタ内の T0IF ビットが 1 となる. 割込みを許可していればこの時点で割込みが発生. この T0IF ビットはプログラムで CLEAR するまで 1 のままを保持する. したがってオーバーフロー処理でこれをゼロクリアしないと次のオーバーフローが分からなくなる.

4. W レジスタの復帰や出力ビットの書き換えにおいて, swapf 命令や addwf 命令を使う理由を述べよ.
   割り込み中のコンテキストの保存に際して, W レジスタや STATUS レジスタなどのレジスタの値を保存する必要があり, SWAPF 命令を使用することで, Z フラグに影響を与えないから.
5. 今まで実験で行った内容以外の PIC を使用した新しい動作の実験テーマや動作・内容を提案せよ. (例) 押すボタン・回数を指定してロックをかける電子錠など
   SN74HC595N などのシフトレジスタを用いて, 7 セグを少ないピン数で光らせる. その際に複数台の 7 セグを光らせるためにダイナミック点灯を用いる. またデータシートを読み回路設計からプログラミングまでのひととおりを, 複数回の実験使うことで学生に主導的に実施させる.
   発展的な内容して, RTC-8564 などのリアルタイムクロックと 4 つの 7 セグを用いたデジタル時計の作成をしてもよいかもしれない.

4 参考文献

- PIC 命令一覧表 (https://www.mlab.im.dendai.ac.jp/~assist/PIC/appendix/instruction/)
- PIC16F84A Data Sheet(https://akizukidenshi.com/download/PIC16F84A.pdf)
- [PIC] 内蔵 EEPROM へアクセスする (http://nanoappli.com/blog/archives/3455)
- 電子工作室 (http://www.picfun.com/pic08.html)