

マイクロコンピュータ 後期期末レポート

電気情報工学科 2 年

E1533 西総一郎

2017 年 2 月 10 日提出

- 光が流れるプログラム (片道バージョン)
- 光が流れるプログラム (往復バージョン)
- パルスモータ 1-2 相励磁プログラム
- 電子サイコロのプログラム
- タイマ割り込み制御プログラム

目 次

1	リスト 5-5(光が流れるプログラム (片道バージョン))	3
1.1	プログラム概要	3
1.2	フローチャート	3
1.3	ソースコード	4
1.4	実行結果・考察	5
2	リスト 5-6(光が流れるプログラム (往復バージョン))	7
2.1	プログラム概要	7
2.2	フローチャート	7
2.3	ソースコード	8
2.4	実行結果・考察	9
2.5	練習問題 5.10	10

1 リスト 5-5(光が流れるプログラム (片道バージョン))

1.1 プログラム概要

8 個ある LED の 1 個を右端や左端から順次点灯することによって、光が流れるように見えるプログラム。

1.2 フローチャート

このプログラムのフローチャートを図 1 に示す。

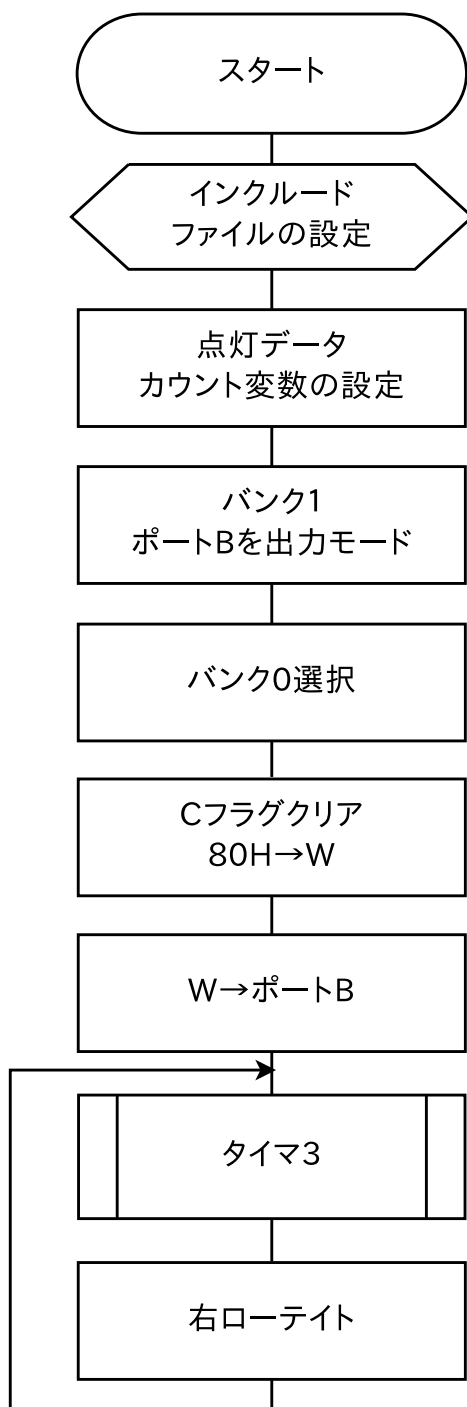


図 1: 光が流れるプログラム (片道バージョン) のフローチャート

1.3 ソースコード

```
1 ; *****
2 ; リスト5.5
3 ; 光が流れるプログラム（片道バージョン）
4 ; （右方向）
5 ; *****
6 LIST P=P16F84A ;使用するPICを指定
7 INCLUDE "P16F84A.INC" ;読み込む設定ファイルを指定
8 ; *****
9 LEDD EQU 80H ;LEDの点灯データの設定
10 CNT1 EQU 0CH ;タイマ1用のカウンタ変数
11 CNT2 EQU 0DH ;タイマ2用のカウンタ変数
12 CNT3 EQU 0EH ;タイマ3用のカウンタ変数
13 ; *****
14 ORG 0 ;プログラムを格納する先頭アドレス
15
16 BSF STATUS,RP0 ;バンク1を選択
17 CLRF TRISB ;ポートBをすべて出力モードに設定
18 BCF STATUS,RP0 ;バンク0を選択
19 BCF STATUS,C ;Cフラグをクリア
20
21 MOVLW LEDD ;点灯データをWレジスタにセット
22 MOVWF PORTB ;点灯データをポートBに出力（LEDが点灯）
23 REPEAT CALL TIMER3 ;0.5秒タイマの呼び出し
24 RRF PORTB,1 ;ポートBを1ビット右にローテイト
25 GOTO REPEAT
26
27 ;止めるまで永遠に繰り返す
28
29 TIMER1 MOVLW D'62' ;0.1ミリ秒タイマサブルーチン
30 MOVWF CNT1
31 LOOP1 NOP
32 DECFSZ CNT1,F
33 GOTO LOOP1
34 RETURN
35
36 TIMER2 MOVLW D'100' ;10ミリ秒タイマサブルーチン
37 MOVLW CNT2
38 LOOP2 NOP
39 CALL TIMER1
40 DECFSZ CNT2,F
41 GOTO LOOP2
42 RETURN
43
44 TIMER3 MOVLW D'50' ;0.5秒タイマサブルーチン
45 MOVWF CNT3
46 LOOP3 NOP
47 CALL TIMER2
48 DECFSZ CNT3,F
49 GOTO LOOP3
50 RETURN
51
52 END ;プログラムの終わり
```

1.4 実行結果・考察

$$80_{16} = 10000000_2$$

$$40_{16} = 01000000_2$$

$$20_{16} = 00100000_2$$

$$10_{16} = 00010000_2$$

$$08_{16} = 00001000_2$$

$$04_{16} = 00000100_2$$

$$02_{16} = 00000010_2$$

$$01_{16} = 00000001_2$$

$$00_{16} = 00000000_2$$

:LED 点灯 :LED 消灯, :C フラグ 1 :C フラグ 0

図 2: 光が流れるプログラム (片道バージョン) の実行結果

図 2 ように、0.5 秒毎に光る場所が右に動き、右端に到達するとすべての LED が消えるタイミングがある。ローテイト (RRF) 命令は 1 ビットずつ右にシフトさせるもので、16 進数において 1 ビット右にシフトさせることは 2 で割ることになる。RRF 命令は C フラグを経由してデータを回転するので端に到達したら一時的に全部が消える瞬間がある。

このプログラムのタイマについて考察してみる。

PIC のクロック周波数は 10MHz なので、1 クロックあたりは

$$\frac{1}{10\text{MHz}} = 0.1\ \mu\text{s}$$

4 クロックで 1 サイクルなので、1 サイクルあたりは

$$0.1\ \mu\text{s} \times 4 = 0.4\ \mu\text{s}$$

TIMER1 のサイクル数は図 3 より、249 サイクルだとわかり、

$$0.4\ \mu\text{s} \times 249 = 99.6\ \mu\text{s} \approx 0.1\ \text{ms}$$

TIMER1 では 0.1 ms 消費される。

この TIMER1 を TIMER2 では 100 回、TIMER2 では 50 回呼び出しているので、

$$0.1\ \text{ms} \times 100 \times 50 = 50\ \text{ms} = 0.5\ \text{s}$$

よって合計で 0.5 s のタイマルーチンであることがわかる。

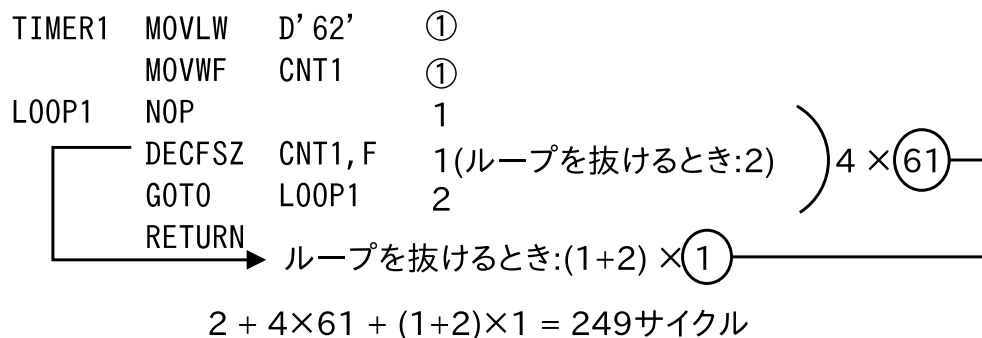


図 3: サイクル数の考え方

ローテイト (RRF) 命令を 0.5 秒のタイマルーチン毎に呼び出すことで、0.5 秒毎に光が動いているように見える。

次に光の移動方向について考察してみる。

このプログラムにおいて、光の移動方向を決めているのは

9 行 目	LEDD	EQU	80H	; LED の点灯データの設定
24 行 目	RRF	PORTB, 1		; ポート B を 1 ビット右にローテイト

この 2 つなので、次のように変更すれば左方向に移動するようになる。

9 行 目	LEDD	EQU	01H	; LED の点灯データの設定
24 行 目	RLF	PORTB, 1		; ポート B を 1 ビット右にローテイト

RLF 命令も RRF 命令と同様に、C フラグを経由してデータを回転するので、端に到達したら一時的に全部が消える。

LED の移動時間 S を調整するには TIMER1 の 0.1 ミリ秒を基準にして、

$$S = 0.1 \text{ ms} \times \text{TIMER2} \times \text{TIMER3}$$

$$\text{TIMER2} \leq 255, \text{TIMER3} \leq 255$$

TIMER2 と TIMER3 でループ回数を決める。

2 リスト 5-6(光が流れるプログラム (往復バージョン))

2.1 プログラム概要

8 個 LED の 1 個を左端や右端から順次点灯していき、端に到達したら逆方向に点灯させることで LED の点灯が、往復して流れるように見えるプログラム。

2.2 フローチャート

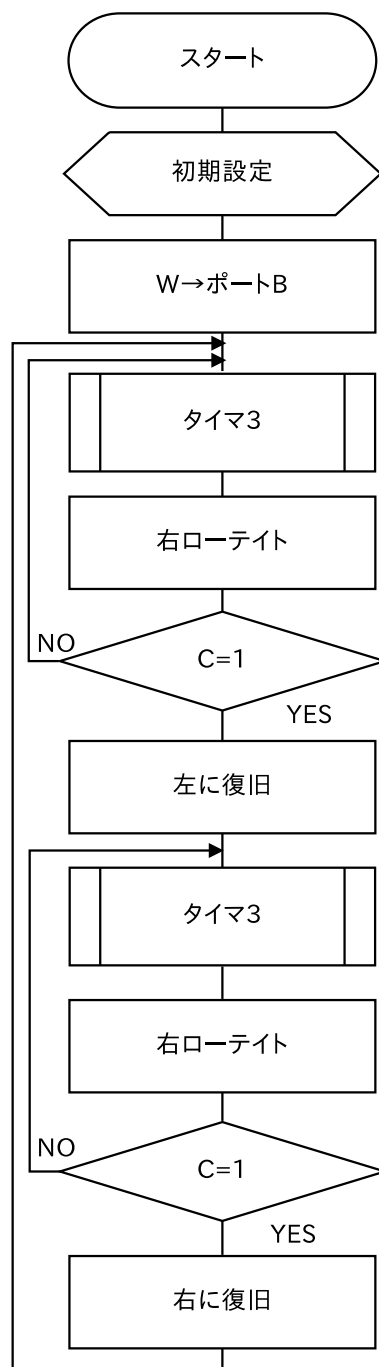


図 4: 光が流れるプログラム (往復バージョン) のフローチャート

2.3 ソースコード

```

1 ; *****
2 ; リスト5.6
3 ; 光が流れるプログラム（往復バージョン）
4 ; （Cフラグで判定バージョン）
5 ; *****
6     LIST      P=P16F84A      ;使用するPICを指定
7     INCLUDE   "P16F84A.INC"  ;読み込む設定ファイルを指定
8 ; *****
9 LEDD      EQU      80H      ;LEDの点灯データの設定
10 CNT1      EQU      0CH      ;タイマ1用のカウンタ変数
11 CNT2      EQU      0DH      ;タイマ2用のカウンタ変数
12 CNT3      EQU      0EH      ;タイマ3用のカウンタ変数
13 ; *****
14     ORG       0              ;プログラムを格納する先頭アドレス
15
16     BSF       STATUS,RP0     ;バンク1を選択
17     CLRF      TRISB          ;ポートBをすべて出力モードに設定
18     BCF       STATUS,RP0     ;バンク0を選択
19     BCF       STATUS,C       ;Cフラグをクリア
20     MOVLW     LEDD           ;点灯データをWレジスタにセット
21     MOVWF     PORTB          ;点灯データをポートBに出力（LEDが点灯）
22
23 RIGHT     CALL    TIMER3      ;0.2秒タイマの呼び出し
24           RRF      PORTB,1     ;ポートBを1ビット右にローテイト
25           BTFSS    STATUS,C    ;Cフラグが1なら次の命令をスキップ
26           GOTO     RIGHT
27
28           RLF      PORTB,1     ;過分ローテイトの復旧
29           RLF      PORTB,1     ;過分ローテイトの復旧
30
31 LEFT      CALL    TIMER3      ;0.2秒タイマの呼び出し
32           RLF      PORTB,1     ;ポートBを1ビット左にローテイト
33           BTFSS    STATUS,C    ;Cフラグが1なら次の命令をスキップ
34           GOTO     LEFT
35
36           RRF      PORTB,1     ;過分ローテイトの復旧
37           RRF      PORTB,1     ;過分ローテイトの復旧
38           GOTO     RIGHT
39
40 TIMER1     MOVLW     D'62'      ;0.1ミリ秒タイマサブルーチン
41           MOVWF     CNT1
42 LOOP1      NOP
43           DECFSZ    CNT1,F
44           GOTO      LOOP1
45           RETURN
46
47 TIMER2     MOVLW     D'100'     ;10ミリ秒タイマサブルーチン
48           MOVWF     CNT2
49 LOOP2      NOP
50           CALL      TIMER1
51           DECFSZ    CNT2,F
52           GOTO      LOOP2
53           RETURN
54
55 TIMER3     MOVLW     D'20'      ;0.2秒タイマサブルーチン
56           MOVWF     CNT3
57 LOOP3      NOP
58           CALL      TIMER2
59           DECFSZ    CNT3,F
60           GOTO      LOOP3
61           RETURN

```


2.4 実行結果・考察

$80_{16} = 10000000_2$	$01_{16} = 00000001_2$
$40_{16} = 01000000_2$	$02_{16} = 00000010_2$
$20_{16} = 00100000_2$	$04_{16} = 00000100_2$
$10_{16} = 00010000_2$	$08_{16} = 00001000_2$
$08_{16} = 00001000_2$	$10_{16} = 00010000_2$
$04_{16} = 00000100_2$	$20_{16} = 00100000_2$
$02_{16} = 00000010_2$	$40_{16} = 01000000_2$
$01_{16} = 00000001_2$	$80_{16} = 10000000_2$
:LED 点灯	:LED 消灯

図 5: 光が流れるプログラム (往復バージョン) の実行結果。左の列のあと右の列が実行される。

図 5 のように左端から、0.2 秒毎に光るところが右に動いていき右端になったら、移動方向を左にして移動する。

ローテイト命令 (RRF, RLF) は、C フラグを含めてシフトするので、光が右端 (0 ビット目) または左端 (7 ビット目) に移動したことを C フラグで判定している。C フラグが 1 の場合はオーバーフローかアンダーフローしているので、図 7 のように過分ローテイトの復旧 (2 ビット復旧) することで、なめらかに移動するように見える。

7 6 5 4 3 2 1 0	C
$01_{16} = 00000001_2$	0 ビット目点灯
$00_{16} = 00000000_2$	C フラグにアンダーフロー
$02_{16} = 00000010_2$	過分ローテイト復旧 (RLF \times 2)
:LED 点灯	:LED 消灯, :C フラグ 1 :C フラグ 0

図 6: 過分ローテイト復旧の例

光の移動方向については、過分ローテイトの復旧を行っているため、LED の点灯データを変えるだけで、移動方向を変えられる。

これを

9 行 目	LEDD	EQU	80H	; 左 端 から 右 方 向 に ス タ ー ト
-------	------	-----	-----	--------------------------

次のように変更すれば、スタート時の移動方向を反対にできる。

9 行 目	LEDD	EQU	01H	; 右 端 から 左 方 向 に ス タ ー ト
-------	------	-----	-----	--------------------------

次に過分ローテイトの復旧がない場合について考えてみる。過分ローテイトの復旧がないと、

7 6 5 4 3 2 1 0	C
$02_{16} = 00000010_2$	1 ビット目点灯
$01_{16} = 00000001_2$	0 ビット目点灯
$00_{16} = 00000000_2$	C フラグにアンダーフロー
$01_{16} = 00000001_2$	0 ビット目点灯
$02_{16} = 00000010_2$	1 ビット目点灯
:LED 点灯	:LED 消灯, :C フラグ 1 :C フラグ 0

図 7: 過分ローテイト復旧がない場合

すべての LED が点灯しない瞬間が生じる。

2.5 練習問題 5.10