工学ゼミⅢ

安全・環境活動に役立つ LED イルミネーションの製作

１．概要

3～5 名の学生グループで安全・環境活動に役立つ LED イルミネーションを作製する。作品のデザイン画や部品リスト、回路図、動作フロー図等は事前に作成し、計画的に作業を行うことが求められる。

２．達成すべき目標

作品に係る資料を事前にまとめ、それに基づいて製作が行える。集団の中で、自身の知識・技術を積極的に応用しながら、実用的でオリジナルな作品を導くことができる。

３．制限条件

電子部品（LED，マイコン，抵抗器，ブレッドボード，配線コード，電池，電池ボックス）は支給されたものを使用すること。外観に使用する部材は 1 グループあたり 1,000 円までの経費（自己負担）を認める。

４．競うこと

発表会では実用性とデザイン性（美しさ）について各学生が評価し、作品に順位を付ける。上位の優れた作品については、学内で展示を行う予定である。

５．その他

作品の製作過程は iPad で写真を撮り、ワークシートの提出や最終発表会で活かすこと。

６．スケジュール

第１回　ガイダンス;サンプル作品の紹介。グループ毎に集まり、担当や作品のコンセプトを決める。
　　担当：リーダー、設計、部品調達、製作（加工/プログラミング）
第２回　設計；作品のデザイン画、部品リスト、回路図、動作フロー図をまとめる。
第３回　製作①；グループ内で作業を分担し、作品の完成を目指す。
第４回　製作②
第５回　製作③　　※30 秒の作品紹介ビデオを作成し、発表会２日前までに Moodle(工学ゼミⅢ)へアップロードする。
第６回　発表会；各グループが作成した作品紹介のビデオを上映する。

７．評価方法　※詳細は Moodle 掲載のルーブリックを参照

作品が当初のコンセプト・計画どおりに完成できたかどうか[10 点]、作品に取り入れたアイディアの独立性と実用性[10 点]、各回のワークシートの内容[4 点×5 回]及び自己評価[7 点]を合わせ、47 点満点で評価する。

８．PIC マイコンとプログラム開発環境

８－１．PIC マイコン

PIC ファミリの 16FXX シリーズは、CPU、演算機能、メモリ、タイマなどの周辺機能を１チップに収めたマイコンである。そのため、数個の部品と電源を接続すれば、センサの波形処理やモータ制御等が容易に行える。Fig.1 は、この PIC マイコンを使用してライントレーサを製作した例であり、Fig.2 はその回路構成を示している。マイコンは、反射型フォトインタラプタ（光センサ）の情報をもとにラインの検出を行い、左右のモータスピードを制御している。

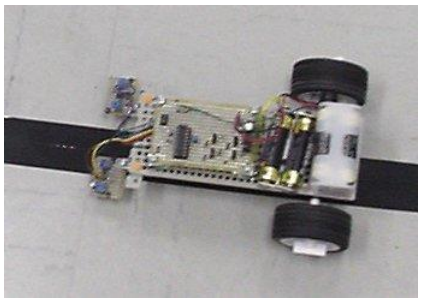


Fig.1 ライントレーサ

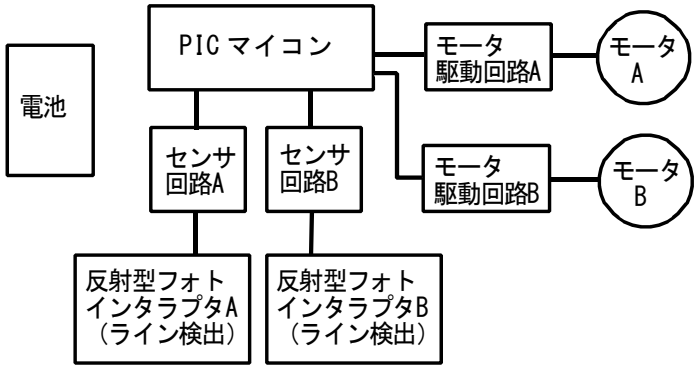


Fig.2　ライントレーサの回路構成

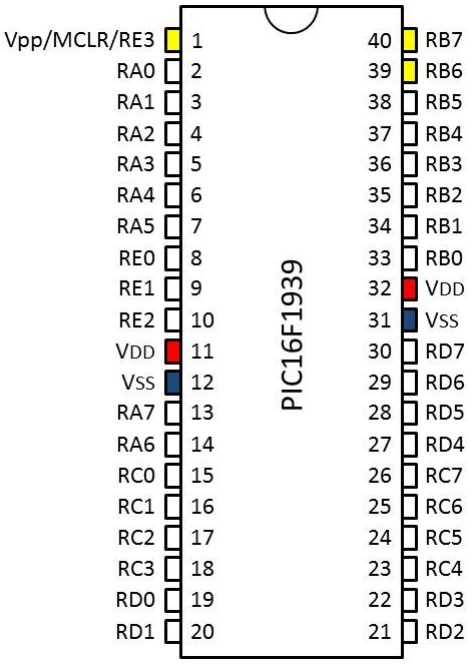


Fig.3 PIC16F1939 のピン配置図

本演習では、PIC16F1939 を使用する。このマイコンは、高精度のオシレータ（クロック）を内蔵し、16k×4 ワードのフラッシュプログラムメモリと 1024 バイトのデータメモリ(RAM)を持っている。

PIC16F1939 のピン配置図を Fig.3 に示す。LED の点灯制御に利用可能な入出力端子は次の 35 箇所である。

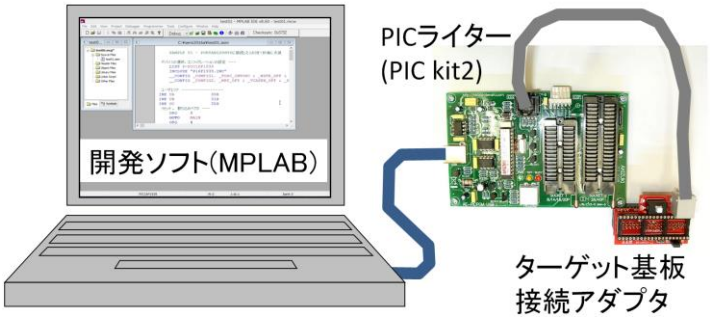
「RA0 から RA7(**PORTA**)、RB0 から RB7(**PORTB**)、RC0 から RC7(**PORTC**)、RD0 から RD7(**PORTD**)及び RE0 から RE2(**PORTE**)」(RE3 は信号出力が出来ない)

VDD は電源（電池）のプラス側、VSS はマイナス側に接続する。（接続は片方のみで良い）また、RE3(1)、RB6(39)及び RB7(40)はプログラム書き込み時に利用される。

８－２．プログラム作成手順

以下の手順で PIC マイコンのプログラムを作成する。

① プログラム設計（動作フロー図の作成）
② コーディング（ソースファイル ***.ASM の作成）
③ アセンブル（オブジェクトファイル → ヘキサファイル ***.HEX の作成）　} **MPLAB** を使用
④ シミュレーション（誤りの検出）
⑤ **PIC ライター**を使用してマイコンにプログラムを書き込む　　**PIKkit 2 Programmer** を使用
⑥ 実機のテスト



※PIC マイコンをライター右側のソケットに挿して書き込みをパソコンで操作する。ターゲット基板接続アダプタは今回は利用しない。

Fig.4 プログラム開発環境

８－３．プログラムの基本

８－３－１．アセンブラ命令

PIC マイコンは全部で 35 個のアセンブラ命令（表 2）を持つ。その他、コーディング時には、他のソースファイルの読み込みやプログラムの開始番地の指定、終わりを示す**擬似命令**を使用してプログラムの記述を行う。

表１．主な擬似命令

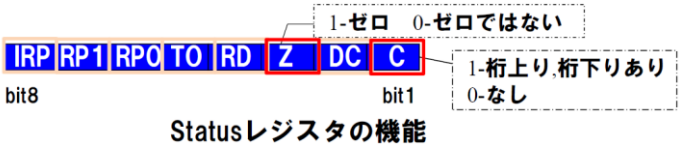
擬似命令	書　式	内　容
__CONFIG	__CONFIG〈式〉	コンフィグレーションビットの設定
END	END	ソースファイルの終わりの宣言
EQU	〈ラベル〉　EQU　〈式〉	定数ラベルの定義
#INCLUDE	#INCLUDE〈ファイル名〉	他のソースファイルの読み込み
ORG	ORG〈式〉	プログラム開始番地の指定

表２．アセンブラ命令

分類	命　令	機　能	影響フラグ	サイクル
バイト処理命令	ADDWF　f , d	加算 W + f → W か f へ格納	C , DC , Z	1
	ANDWF　f , d	論理積 W AND f → W か f へ格納	Z	1
	CLRF　f	f をゼロクリア	Z	1
	CLRW	W を ゼロクリア	Z	1
	COMF　f , d	f の0 , 1 反転 → W か fへ格納	Z	1
	DECF　f , d	f − 1 → W か f へ格納	Z	1
	DECFSZ　f , d	f − 1 → W か f　結果ゼロなら次命令スキップ		1(2)
	INCF　f , d	f + 1 → W か f へ格納	Z	1
	INCFSZ　f , d	f + 1 → W か f　結果ゼロなら次命令スキップ		1(2)
	IORWF　f , d	論理和 W OR f → W か f へ格納	Z	1
	MOVF　f , d	移動 f から W または f 自身へ格納	Z	1
	MOVWF　f	移動 W から f へ格納		1
	NOP	何もしない		1
	RLF　f , d	1ビット 左へシフト → W か f へ格納	C	1
	RRF　f , d	1ビット 右へシフト → W か f へ格納	C	1
	SUBWF　f , d	減算 f − W → W か f へ格納	C , DC , Z	1
	SWAPF　f , d	f の 上位と下位を入れ替え → W か f へ格納		1
	XORWF　f , d	排他的論理和 W XOR f → W か f へ格納	Z	1
ビット処理命令	BCF　f , b	f の bビット 目をゼロにクリアする		1
	BSF　f , b	f の bビット 目を1にセットする。		1
	BTFSC　f , b	f の bビット 目がゼロだったら次命令スキップ		1(2)
	BTFSS　f , b	f の bビット 目が1だったら次命令スキップ		1(2)
リテラル処理命令	ADDLW　k	定数加算 W + k → W へ格納	C , DC , Z	1
	ANDLW　k	定数論理積 W AND k → W へ格納	Z	1
	IORLW　k	定数論理和 W OR k → W へ格納	Z	1
	MOVLW　k	定数移動 k → W へ格納		1
	SUBLW　k	定数減算 k − W → W へ格納	C , DC , Z	1
	XORLW　k	排他的論理和 W XOR k → W へ格納	Z	1
ジャンプ命令	CALL　k	サブルーチン k へジャンプ		2
	GOTO　k	k 番地へジャンプ		2
	RETFIE	割り込み許可で戻る		2
	RETLW　k	W に k を格納して戻る		2
	RETURN	サブルーチンから戻る		2
他	CLRWDT	ウォッチドックタイマクリア		1
	SLEEP	スリープモードにする		1

*1 オペランド部の **f** は任意のレジスタ，**d** は格納先の選択（0 : W レジスタ，１： f の指定レジスタ），**b** はビット位置（0～7），**k** はリテラル（定数データ）を表す。

*2 影響フラグに C, DC, Z などの記載があるものは、その命令の実行によって、STATUS(ステータス) レジスタ内の対応するビットが影響を受けることを意味する。



Statusレジスタの機能

８－３－２．プログラムの形式

```
;       SAMPLE 01 - PORTA から PORTD に接続した LED を１秒毎に点滅
;
;---  デバイスの選択，コンフィグレーションの設定 ---
        LIST P=PIC16F1939
        INCLUDE "P16F1939.INC"
      __CONFIG _CONFIG1, _FOSC_INTOSC & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF
& _BOREN_OFF & _CLKOUTEN_OFF & _IESO_OFF & _FCMEN_OFF
      __CONFIG _CONFIG2, _WRT_OFF & _VCAPEN_OFF & _PLLEN_ON & _STVREN_OFF & _BORV_19 & _LVP_OFF

;--- ユーザエリア ----------------
#DEFINE UA      30H
#DEFINE UB      31H
#DEFINE UC      32H
;--- リセット，割り込みベクタ ---
        ORG     0
        GOTO    MAIN
        ORG     4
        GOTO    MAIN
;========= メインルーチン ==============
MAIN
        BCF     INTCON,7        ;割り込み不許可
        BSF     BSR,BSR0        ;バンク１に切り替え
        MOVLW   B'00000000'
        MOVWF   TRISA           ;ポート A のデータ方向（入出力）設定
        MOVLW   B'00000000'
        MOVWF   TRISB           ;ポート B のデータ方向（入出力）設定
        MOVLW   B'00000000'
        MOVWF   TRISC           ;ポート C のデータ方向（入出力）設定
        MOVLW   B'00000000'
        MOVWF   TRISD           ;ポート D のデータ方向（入出力）設定
        MOVLW   B'00000000'
        MOVWF   TRISE           ;ポート E のデータ方向（入出力）設定
        MOVLW   B'01101010'
        MOVWF   OSCCON          ;内部発生クロック 4MHz 選択
        ;-------------------------
        MOVLW   080H            ;オプションレジスタの設定
        MOVWF   OPTION_REG
        BCF     BSR,BSR0        ;バンク 0 に切り替え
        NOP

LOOP
        MOVLW   B'11111111'
```

```
        MOVWF   PORTA
        MOVWF   PORTB
        MOVWF   PORTC
        MOVWF   PORTD
        MOVWF   PORTE
        CALL    WAIT0

        MOVLW   B'00000000'
        MOVWF   PORTA
        MOVWF   PORTB
        MOVWF   PORTC
        MOVWF   PORTD
        MOVWF   PORTE
        CALL    WAIT0
        GOTO    LOOP
;------------------------
;     1SEC                      ;１秒カウント(待機)するためのサブルーチン
;------------------------
WAIT0
        MOVLW   D'10'             ;
        MOVWF   UC                ;UC=10 とする
WAIT0A
        CALL    WAIT1             ;WAIT1 をサブルーチンコール
        DECFSZ  UC,1              ;UC の値を-1 し、ゼロならば GOTO 命令をスキップ
        GOTO    WAIT0A                  ;ラベル WAIT0 へジャンプ
        RETURN                          ;
;------------------------
;     100mSEC                   ;100 ミリ秒カウントするサブルーチン
;------------------------
WAIT1
        MOVLW   D'150'
        MOVWF   UA                ;UA=150
WAIT1A
        MOVLW   D'223'
        MOVWF   UB                ;UB=223
WAIT1B
        DECFSZ  UB,1              ;UB を-1 し、ゼロならば GOTO 命令をスキップ
        GOTO    WAIT1B                  ;ラベル WAIT1B へジャンプ
        DECFSZ  UA,1              ;UA を-1 し、ゼロならば GOTO 命令をスキップ
        GOTO    WAIT1A            ;ラベル WAIT1A へジャンプ
        RETURN

        END                       ;ソースファイルの終わりを宣言;
```

※サンプルプログラムの補足説明

1）数値データの表記

10 進	D’10’	;10
16 進	H’10’ もしくは 10H	;16
2 進	B’00001010’	;10

2）プログラムメモリとデータメモリ領域

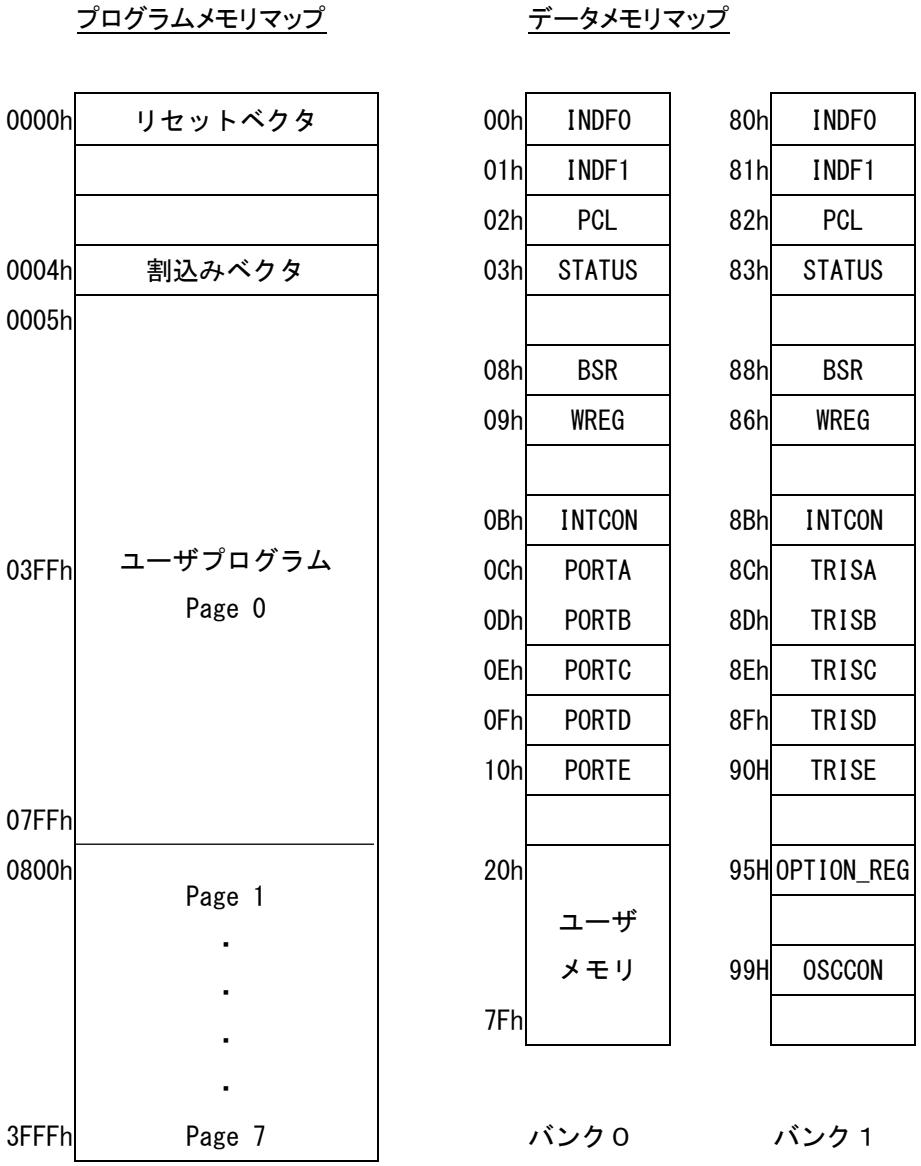


Fig.5　プログラムメモリマップとデータメモリマップ

データメモリは、128 バイト単位で分けられており、それをバンクと呼ぶ。バンクの移動は BSR レジスタの BSR4 から BSR0 ビットを設定することで行われ、全てのビットが 0 でバンク 0, BSR0 のみ 1 でバンク 1 の状態になる。なお、ユーザがプログラムで利用可能なメモリは、バンク 0 においてアドレス 20h から 7Fh の範囲にある

３）コンフィグレーションの設定

コンフィグレーションビットは、PIC マイコンのハードウェアの動作を決定するもので、プログラム同様に PIC ライタを使用して特定のメモリに書き込みを行う。本演習では、上記のサンプルと同じ設定とすることを薦める。

```
__CONFIG _CONFIG1, _FOSC_INTOSC & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_OFF & _CLKOUTEN_OFF & _IESO_OFF & _FCMEN_OFF
```

（内蔵クロック利用で外部出力しない & ウォッチドックタイマ無効 & パワーアップタイマ無効 & MCLR 端子無効 & コードプロテクトしない & データ EEPROM プロテクトしない など）

```
__CONFIG _CONFIG2, _WRT_OFF & _VCAPEN_OFF & _PLLEN_ON & _STVREN_OFF & _BORV_19 & _LVP_OFF
```

４）入出力端子（PORTA から PORTE）におけるデータ方向の設定

入出力端子に LED を接続し点滅動作を行うには、プログラムの初期段階でデータ方向を“出力”に設定しておく必要がある。

PORTA のデータ方向設定レジスタ：TRISA（85H）

bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

出力－0，　入力－1

同様に

PORTB のデータ方向設定レジスタ：TRISB（86H）
PORTC のデータ方向設定レジスタ：TRISC（86H）
PORTD のデータ方向設定レジスタ：TRISD（86H）
PORTE のデータ方向設定レジスタ：TRISE（86H）

＜参考書・ホームページ＞

１．角山正博，佐藤栄一；コンピュータの基礎，青山社，2013.
２．http://www.picfun.com/ ，電子工作の実験室，後閑.
（PIC マイコンの機能やプログラミングについて解かりやすく解説）
３．http://www.microchip.co.jp/ ，マイクロチップテクノロジージャパン.
（MPLAB や PIC16F1939 などのデータシートをダウンロード可能）
４．https://esato.net/ex/micom/ ，PIC マイコンによる LED イルミネーション製作，佐藤.
（このテーマのサポート HP）