

## 실험 2. LED 동작시키기

### ■ 예비문제 2

1) PIC16F876의 memory map의 내용을 이해합시다. 특히 초기 시작 번지가 어디이며 사용 가능한 전체 memory 영역이 어디까지인가 확인합시다.

2) PIC16F876의 register file map의 내용을 이해합시다. Map 중에서 이름이 있는 영역과 이름이 없는 영역이 있는데, 이것의 차이는 무엇인지 이해해 봅시다.

3) 실험에서 사용하는 프로그램에 사용된 명령어의 의미를 부록에 있는 명령어 설명을 보고 이해합시다.

4) 다음 프로그램의 의미를 이해해 봅시다.

```
LIST          P=16F876
STATUS      EQU      03H
W           EQU      B'0'
F           EQU      .1
;
ORG         0
MOVLW      33
MOVWF      STATUS
GOTO      SIJAK
.
.
SIJAK      MOVLW      2FH
          BSF        W,F
.
.
END
```

---

## ■ 실 험

다음 프로그램을 수행시켜 보고 동작을 확인합시다.

- 1) 다음 프로그램을 작성하여 TEST1.ASM 으로 저장한다.

```

; STANDARD HEADER FILE
        PROCESSOR      16F876
; --- REGISTER FILES 선언 -----
; BANK 0
INDF      EQU      00H
TMR0      EQU      01H
PCL       EQU      02H
STATUS    EQU      03H
FSR       EQU      04H
PORTA     EQU      05H
PORTB     EQU      06H
EEDATA    EQU      08H
EEADR     EQU      09H
PCLATH    EQU      0AH
INTCON    EQU      0BH
; BANK 1
OPTINOR   EQU      81H
TRISA     EQU      85H
TRISB     EQU      86H
EECON1    EQU      88H
EECON2    EQU      89H
ADCON1    EQU      9FH
; --- STATUS BITS 선언 -----
IRP       EQU      7
RP1       EQU      6
RP0       EQU      5
NOT_TO    EQU      4

```

---

```

NOT_PD    EQU        3
ZF         EQU        2    ; ZERO FLAG BIT
DC         EQU        1    ; DIGIT CARRY/BORROW BIT
CF         EQU        0    ; CARRY/BORROW FLAG BIT

```

```

; -- INTCON BITS 선언 -----
; -- OPTION BITS 선언 -----

```

```

W          EQU        B'0' ; W 변수를 0으로 선언
F          EQU        .1   ; F 변수를 1로 선언
; MAIN PROGRAM
    ORG    0000
    BSF    STATUS,RP0 ; BANK를 1로 변경함
    MOVLW  B'00000000' ;
    MOVWF  TRISA
    MOVLW  B'00001111'
    MOVWF  TRISC
    MOVLW  B'00000111' ; PORTA를 DIGITAL I/O로사용
    MOVWF  ADCON1
    BCF    STATUS,RP0 ; BANK를 0으로 변경
;
    MOVLW  00H
    MOVWF  PORTA ; PORTA 초기화
LOOP INCF  PORTA,F ; PORTA를 1씩 증가시킴
    GOTO   LOOP

    END

```

- 2) MOVF PORTA,W와 MOVF PORTA,F를 부록의 명령어를 찾아보면, MOVE f to d로 설명되어 있다. 즉 f를 d로 옮겨라는 명령어이다. 여기서 f는 file register이며, register map에 있는 f 번지의 RAM으로 된 기억장치에 기억된 내용을 의미하며, d는 이를 저장하는 위치를 지정하는 변수이다. 그런데, 설명에 따르면 d의 값에 따라서 d가 '0'이면 W register에
-

저장하고, '1'이면 f인 file register에 저장하라고 되어 있다. 따라서 위 명령어 MOVF PORTA,W를 번역하면 PORTA는 05번지를 지정하고, W는 W EQU 0의 선언으로 W=0이므로 MOVF 05, 0이 되고, 따라서 05번지의 내용을 읽어서 ALU에 있는 W-register(working register)에 기억시켜라는 명령어가 된다.

동일한 의미로 MOVF PORTA,F는 MOVF 05, 1이 되며, 이는 05번지의 내용을 읽어서 05번지에 기억시켜라는 명령어가 된다. 즉 동일한 기억장소에 있는 내용을 읽어서 동일한 장소에 넣어라는 의미가 되어 필요없거나 잘못된 명령어로 이해할 수 있으나, 이 과정에서 읽는 내용이 '0' 인가 아닌가를 검색할 수 있도록 하는 기능이 수행된다. 자세한 것은 명령어의 STATUS라는 항을 이해하면 된다.

- 3) 이를 assembler를 수행시켜 보시다. 그러면 error가 발생할 것입니다. Error의 의미를 이해하고 수정해 보시다.
  - 4) Error가 없을 때, 화면에 나타나는 내용들을 자세히 확인하여 프로그램이 어떻게 변환되었는가를 이해합시다. 그리고 Step into를 선택하여 한 명령어씩 수행시키면서 화면 내용들을 관찰합시다. 특히 View 창의 **Special function register 영역의 내용과 File register window 영역의 내용**에 주목합시다.
- ☞ 붉은 색으로 변하는 부분은 이전 상태와 다른 값을 나타내는 영역이므로 주의 깊게 따라가 볼 것!
- 5) 4)의 결과로부터 작성한 프로그램 각 줄에 대하여 의미와 내용을 설명하십시오.
  - 6) 위 프로그램을 정상 속도로 수행시키면 (RUN를 선택하면 됨) 화면에 변화가 있는가? 없다면 이는 무엇을 의미하는가를 설명해 보시오. 그리고 프로그램 동작 시 RUN과 STEP과의 차이를 설명해 보시오.
  - 7) 위 실험은 하드웨어 없이 프로그램만의 동작을 확인한 것이다. 그러나 PORTA와 PORTB는 마이크로프로세서의 내부 신호를 외부와 연결시켜

주는 I/O 장치(port)로 외부에 회로를 연결하면 마이크로프로세서의 내부 동작 상태를 알 수 있거나, 외부 신호를 마이크로프로세서 내부로 넣어 줄 수 있다. 따라서 MOVWF PORTA는 마이크로프로세서 내의 ALU에 있는 W-reg.에 기억된 내용을 PORTA로 출력(기억)하라는 명령어가 되며, 이는 외부에 LED를 연결하여 확인할 수 있다. 또 INCF PORTA,F 명령어는 PORT에 들어있는 내용을 1씩 증가시켜 저장하라는 명령이며, GOTO LOOP 명령어는 이를 반복하라는 의미이므로 PORTA로 출력되는 값이 매 LOOP 마다 1씩 증가됨을 나타낸다. 이를 실험으로 확인하기 위하여 다음 회로를 구성하고(점선으로 표시된 저항 LED는 실험 데모보드에 이미 구성되어 있는 회로이므로 PIC16F876 pin과 데모보드 pin 사이만 선으로 연결하면 된다. 회로가 정상적으로 연결되었는가를 확인하고 전원을 on하고, 프로그램을 down load하여 step-into으로 동작시키면서 명령어에 따른 LED의 상태를 확인하시오. 그리고 RUN으로 동작시키면 어떠한가도 확인하시오.

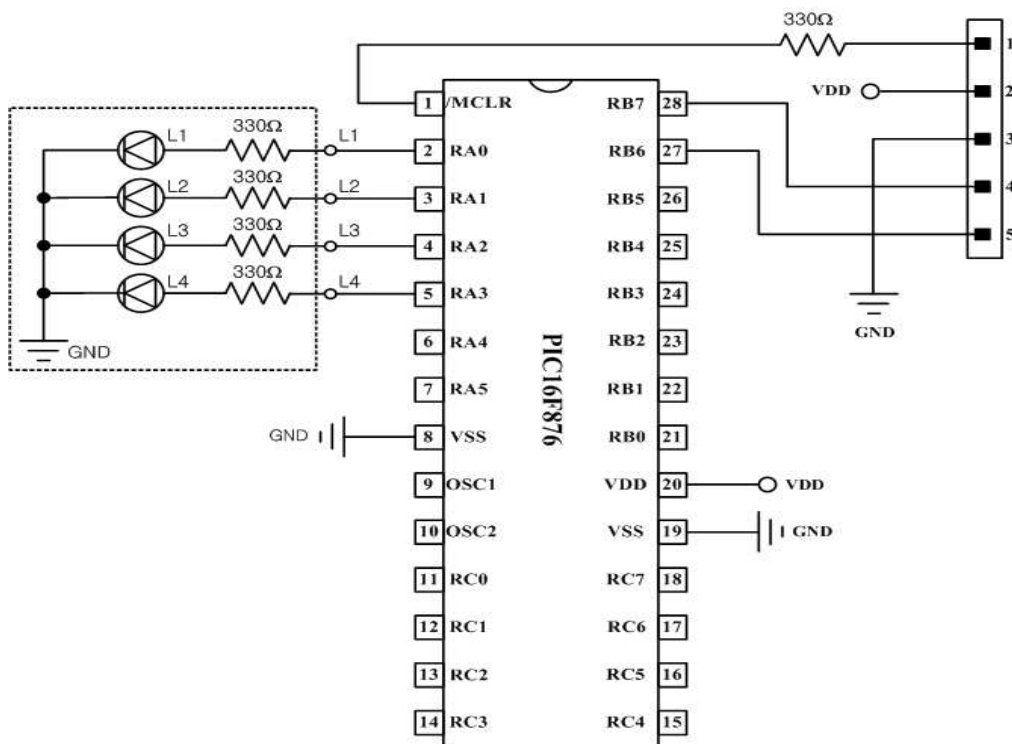


그림 E1-1. LED 구동회로

☞ 참고: 컴퓨터는 2진 숫자를 다루므로 PORTA를 0에서부터 1씩 증가시키면, LED는 0000에서 0001,0010,0011,0100 .. 1111,0000으로 반복된다. 즉 hex code로는 00H(00000000)에서 FFH(11111111)까지 변하며, 이중 low nibble(bit3,2,1,0) 값이 LED로 출력된다.

8) 위 회로에 다음의 PUSH 스위치 회로를 추가하고 프로그램을 약간 변경하여 S/W의 상태를 읽어서 led로 출력하는 프로그램을 작성하려고 한다. 먼저 프로그램을 어떻게 수정해야하는가? 이때 스위치 P1, P2, P3, P4는 각각 LED L1, L2, L3, L4를 제어하도록 하시오.

☞ Hint : INCf PORTA,F 명령어 대신 PORTC를 읽어오는 명령어와 PORTA로 출력하는 명령어를 추가하면 됩니다.

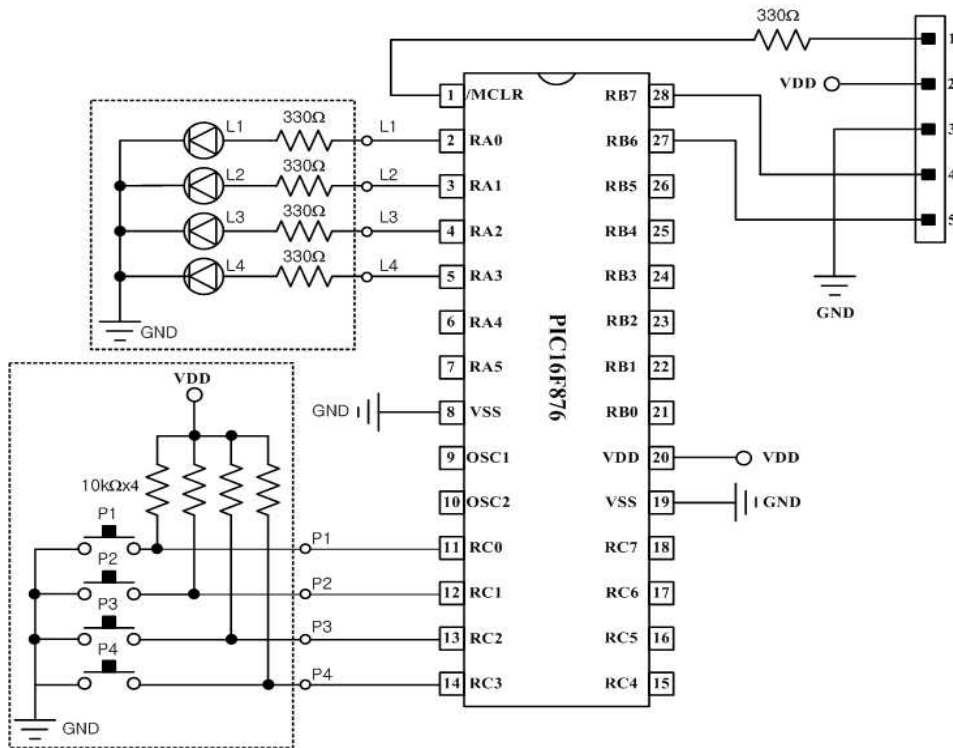


그림 E1-2. Push s/w 와 LED 구동회로

- 9) 수정된 프로그램을 수행시켜 W-reg의 내용이 스위치의 상태에 따라서 어떻게 변하는가를 확인하여(step into 명령으로 프로그램을 수행시켜야만 변함) 명령어 'MOVF PORTA, W' 의 의미를 이해하도록 합시다.
- 10) 9) 실험에서는 스위치를 누를 경우에 불이 꺼지나, 이를 회로의 변경 없이 program의 변경으로 반대의 동작을 시키려고 한다. 명령어를 추가하여 구현해보시오.

☞ Hint : 여러 가지 방법이 있으나, 명령어 1개 만을 추가하여 만들 수 있습니다. 그리고 기존 프로그램을 변경하면, 프로그램 길이가 늘어나지(추가되는 명령어 없이) 않게도 할 수가 있습니다.

☞ 주의점 : 실험에서 작성하는 program은 앞으로 계속 수정 보완되면서 사용해야 하므로 각자 자기 directory에서만 사용하도록 하고 다른 실험 반에 들어가 program의 내용을 복사하거나 변경하는 일이 없도록 합시다. 처음에 조금 편하게 하려고 이해하지 않고 다른 사람의 program을 그냥 가져올 경우 프로그램 길이가 길어지면 남의 프로그램은 이해할 수가 없게 됩니다.

## ■ HOME WORK 2

- 1) Directive 가 어떤 것이 있으며, 의미를 알아봅시다.
- 2) 여러분이 시계를 설계 제작하려고 한다. 외형적인 것은 제외하고 기본 동작을 어떻게 구현 해야되는가를 구체적인 흐름으로 설계해 봅시다.
- 3) ALU와 직접 연관되는 명령들이 무엇이며, 어떻게 동작되는가를 설명합시다.
- 4) 마이크로프로세서는 한 선에 여러 개의 신호가 이동하게 된다. 이렇게 하려면 다른 논리소자의 출력을 공통으로 연결하여야 한다. 이 경우 합선되지 않고 정상적으로 신호가 전달되기 위한 방법은 무엇인가 설명하시오.
- 5) File register 중에 STATUS라는 것이 있다. 이것의 각 bit의 의미와 사용 방법을 이해합시다. 그리고 BTFSS, BTFSC의 사용 방법을 이해합시다.
- 6) Register files에서 왜 bank가 필요한가를 설명하고, 이것을 관리하는 방법은 무엇인가를 설명하시오.