

# 10조 Term Project 결과보고서

담당 교수 : 김종화 교수님  
제출 일시 : 2018년 6월 21일  
10조 : 이준영 (2014104272)  
위세훈 (2014104210)

## 목 차

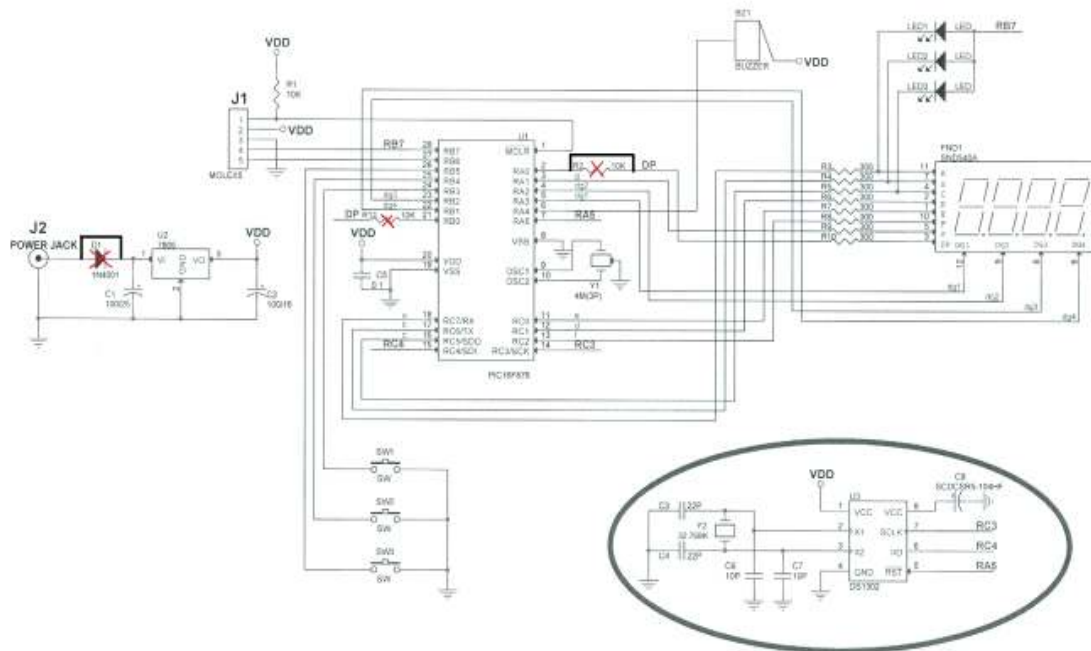
1. 과제명
2. 구현내용 요약
3. 인터페이스 회로도 및 입출력 할당 테이블
4. 구현 기능 상세 설명
5. 소스코드
6. 조원 별 분담 내용
7. 결론 및 고찰

## 1. 과제명 : PIC16F876A & 7-SEG를 이용한 알람시계

### 2. 구현내용 요약

기능	내용
시계(시/분/초)	24시 시간을 (시, 분) 표시, (초) 표시로 확인 할 수 있다. 다른 기능을 실행하더라도 시간은 계속 카운트 되어 동작한다.
시간 설정	스위치를 이용하여 현재 시간을 조정할 수 있다. 시간 설정하기 원하는 시간 단위(DISP 자리)를 구분하기 위해 dot가 0.5초 마다 점멸한다.
알람 설정	알람시간을 시/분 단위로 조정할 수 있다. 시간 설정과 마찬가지로 시간 단위(DISP 자리)를 구분하기 위해 dot가 빠르게 점멸한다. 알람을 설정하면 현재시간을 항상 알람시간과 비교하여 시/분/초가 같아지는 순간 부저가 6초 동안 울리고 스위치3으로도 끌 수 있다.
Switch 1	모드를 변경하는 스위치이다. MODE 1 : 시/분 시계 MODE 2 : 초 시계 MODE 3 : 시간 설정 MODE 4 : 알람 설정
Switch 2	MODE 3, MODE 4 :시간을 설정 할 때 시계의 시간 단위(DISP 자리)를 이동시킨다. (DISP0 ->DISP1 -> DISP2 -> DISP3 ->DISP0 ....)
Switch 3	MODE 1, MODE 2 :알람이 울리고 있는 경우 누르면 알람을 수동으로 끌 수 있다. MODE 3, MODE 4 :현재 위치한 DISP의 숫자 표시를 1씩 증가시켜 원하는 시간을 설정한다. 만약 하위 단위의 시간이 9일 때 누르면 0이 되고 상위 단위 시간이 1 증가한다.

### 3. 인터페이스 회로도 및 입출력 할당 테이블



SCANTIMERBD-3.DSN(2015.04.21)

PORT A		PORT B		PORT C	
RA0	7-SEGMENT DP	RB0	X	RC0	7-SEGMENT E
RA1	7-SEGMENT G	RB1	7-SEGMENT DG4	RC1	7-SEGMENT D
RA2	7-SEGMENT DG2	RB2	7-SEGMENT DG3	RC2	7-SEGMENT F
RA3	7-SEGMENT DG1	RB3	SWITCH1	RC3	X
RA4	BUZZER	RB4	SWITCH2	RC4	X
RA5	X	RB5	SWITCH3	RC5	7-SEGMENT C
		RB6	MOLEX5에 연결	RC6	7-SEGMENT B
		RB7	MOLEX4에 연결	RC7	7-SEGMENT A

## 4. 구현기능 상세 설명

### - 변수 설명 -

W_TEMP	: 인터럽트시 이전의 W 값을 저장하기 위한 변수
STATUS_TEMP	: 인터럽트시 이전의 STATUS 값을 저장하기 위한 변수
INT_CNT	: 내부시간 확인을 위한 변수
DISP_CNT	: 시분시계, 알람시계의 DISPLAY ROUTINE 카운트 변수
SEC_CNT	: 초시계의 DISPLAY ROUTINE 카운트 변수
MOD_CNT	: 모드 선택 카운트 변수
CHANGE_CNT	: 시간조정모드에서 DISP 위치를 가지고 있는 카운트 변수 (0: DISP3, 1: DISP2, 2: DISP1, 3: DISP0)
ALARM_CNT	: 알람모드에서 DISP 위치를 가지고 있는 카운트 변수 (0: DISP3, 1: DISP2, 2: DISP1, 3: DISP0)
D_1SEC	: 1초 단위 변수
D_10SEC	: 10초 단위 변수
D_1MIN	: 1분 단위 변수
D_10MIN	: 10분 단위 변수
D_1HOUR	: 1시간 단위 변수
D_10HOUR	: 10시간 단위 변수
D_1SEC_AL	: 알람 1초 단위 변수
D_10SEC_AL	: 알람 10초 단위 변수
D_1MIN_AL	: 알람 1분 단위 변수
D_10MIN_AL	: 알람 10분 단위 변수
D_1HOUR_AL	: 알람 1시간 단위 변수
D_10HOUR_AL	: 알람 10시간 단위 변수
TEMP	: 변수 값을 잠시 저장하기 위한 변수
DBUF0	: 딜레이 변수
DBUF1	: 딜레이 변수
DBUF2	: 딜레이 변수

### - LABEL 설명 -

[ ORG 0004 ]

인터럽트 발생 시 동작할 내용

[ START\_UP ]

회로 설정에 맞게 입/출력 설정, 인터럽트 시간 설정

[ MAIN ]

변수 초기화

[ LOOP ]

스위치1 체크 -> 눌린 경우 MOD\_CHK로 이동

내부시간 확인(INT\_CNT .244 -> 1초 : 인터럽트 2번 발생 시 1증가하기 때문에)

1초 지나면 LOOP1으로 이동

[ LOOP1 ]

시간 변수를 0FH로 masking

1초 마다 시간 증가 및 시간 단위 체크

24시가 되는 경우 TIME\_24로 이동하여 시간 변수 초기화

매 초마다 알람 확인을 위해 ALARM\_CHECK로 이동

---

[MOD\_CHK]

MOD\_CNT 증가 (모드 4개로 분기(0,1,2,3) -> 4가되면 다시 0으로)

0 -> LOOP : 시분 시계

1 -> LOOP : 초 시계 ( 시분 시계, 초시계는 MOD에서 구분)

2 -> CHANGE : 시분 시간 설정

3 -> AL\_LOOP : 알람 시간 설정

[ MOD ]

DISP 1~4 전체 OFF

인터럽트 안에서 MOD\_CNT를 확인하여 해당 위치로 이동

0 -> HM\_DISP : 시분 출력

1 -> SEC\_DISP : 초 출력

3 -> AL\_DISP : 알람 설정 시분 출력

---

MODE2 시분 시간 설정

[ CHANGE ]

0.5초 마다 DOT 점멸을 위해 DOT\_ON, DOT\_OFF로 이동

[ DOT\_ON ]

현재 DISP 위치의 DOT가 켜지도록 D\_ON1~4로 분기시키고

해당 위치의 시간 변수에 10H를 masking하고 DOT로 이동

[ DOT\_OFF ]

현재 DISP 위치의 DOT가 꺼지도록 D\_OFF1~4로 분기시키고

해당 위치의 시간 변수에 0FH를 masking하여 DOT로 이동

[ DOT ]

SW1을 누르는 경우 MOD\_CHK로 이동

SW2를 누르는 경우 MOV\_DIGIT을 CALL (시간 설정할 DISP 위치를 변경)

SW3을 누르는 경우 INC\_NUM로 이동 (현재 DISP 위치의 시간 1씩 증가)

시간 변경 중에도 1초가 지나는 것을 반영하기 위해 INT\_CNT .244 확인 후 1초 증가

[ MOV\_DIGIT ]

CHANGE\_CNT 증가

시간 변수를 0FH로 masking (10H로 마스킹 된 경우를 대비해서) 후 리턴

[ INC\_NUM ]

약간의 딜레이

현재 DISP 위치에 해당하는 INC\_10HOUR ~ INC\_1MIN 으로 분기

INC\_10HOUR ~ INC\_1MIN 에서 시간 변수 값 증가 후 INC\_CHECK로 이동

[ INC\_CHECK ]

시간이 증가되었으므로 시간 단위 체크

다시 CHANGE로 이동해서 위의 과정 반복

-----  
MODE3 알람 설정

[ AL\_LOOP ]

약 0.125초마다 DOT 점멸을 위해 AL\_DOT\_ON, AL\_DOT\_OFF로 이동

[ AL\_DOT\_ON ]

현재 DISP 위치의 DOT가 켜지도록 AL\_D\_ON1~4로 분기시키고

해당 위치의 시간 변수에 10H를 masking하고 AL\_DOT로 이동

[ AL\_DOT\_OFF ]

현재 DISP 위치의 DOT가 꺼지도록 AL\_D\_OFF1~4로 분기시키고

해당 위치의 시간 변수에 0FH를 masking하여 AL\_DOT로 이동

[ AL\_DOT ]

SW1을 누르는 경우 MOD\_CHK로 이동

SW2를 누르는 경우 MOV\_DIGIT\_AL을 CALL (알람 설정할 DISP 위치를 변경)

SW3을 누르는 경우 INC\_NUM\_AL로 이동 (현재 DISP 위치의 시간 1씩 증가)

알람 설정 중에도 1초가 지나는 것을 반영하기 위해 INT\_CNT 확인 후 LOOP1 이동

[ MOV\_DIGIT\_AL ]

ALARM\_CNT 증가

시간 변수를 0FH로 masking (10H로 마스킹 된 경우를 대비해서) 후 리턴

[ INC\_NUM\_AL ]

약간의 딜레이

현재 DISP 위치에 해당하는 시간인 INC\_10HOUR\_AL ~ INC\_1MIN\_AL 으로 분기

INC\_10HOUR\_AL ~ INC\_1MIN\_AL 에서 시간 변수 값 증가 후 INC\_CHECK\_AL로 이동

[ INC\_CHECK\_AL ]

시간이 증가되었으므로 시간 단위 체크

다시 AL\_LOOP로 이동해서 위의 과정 반복

[ ALARM\_CHECK ]

현재 시간과 알람 시간을 비교하여

시간이 같지 않으면 NO로 이동, 시간이 같으면 YES로 이동

[ NO ]

BUZZER OFF 후 MODE3인 경우 AL\_LOOP로, 그 외의 경우 LOOP로 이동

[ YES ]

BUZZER ON 후 SW3을 누르는 경우 BUZZER OFF

SW3을 누르지 않는 경우 알람이 울리다가 6초 지나면 BUZZER OFF

(부저가 울리는 6초간 시간이 흐르는 것을 반영하기 위해 AL\_DELAY 이용)

MODE3인 경우 AL\_LOOP로, 그 외의 경우 LOOP로 이동

-----  
[ HM\_DISP ] [ SEC\_DISP ] [ AL\_DISP ]

7-SEGMENT에 출력할 DISP를 선택하기 위한 루틴

[ DISP~ ] [ S\_DISP~ ] [ AL\_DISP~ ]

각 위치에 해당하는 시간 변수의 숫자를 7-segment에 출력하는 부분

## 5. 소스코드

```
PROCESSOR 16F876A
#include <P16F876A.INC>
```

VARIABLE	W_TEMP	=	20H
VARIABLE	STATUS_TEMP	=	21H
VARIABLE	INT_CNT	=	22H
VARIABLE	DISP_CNT	=	23H
VARIABLE	D_1SEC	=	24H
VARIABLE	D_10SEC	=	25H
VARIABLE	D_1MIN	=	26H
VARIABLE	D_10MIN	=	27H
VARIABLE	D_1HOUR	=	28H
VARIABLE	D_10HOUR	=	29H
VARIABLE	SEC_CNT	=	2AH
VARIABLE	MOD_CNT	=	2BH
VARIABLE	DBUF1	=	2CH
VARIABLE	DBUF2	=	2DH
VARIABLE	CHANGE_CNT	=	2FH
VARIABLE	D_1SEC_AL	=	30H
VARIABLE	D_10SEC_AL	=	31H
VARIABLE	D_1MIN_AL	=	32H
VARIABLE	D_10MIN_AL	=	33H
VARIABLE	D_1HOUR_AL	=	34H
VARIABLE	D_10HOUR_AL	=	35H
VARIABLE	ALARM_CNT	=	36H
VARIABLE	DBUF0	=	37H
VARIABLE	TEMP	=	38H

-----

ORG	0000	
GOTO	START_UP	
ORG	4	
MOVWF	W_TEMP	; 현재 사용되고 있는 W REG. 저장
SWAPF	STATUS, W	
MOVWF	STATUS_TEMP	
MOVF	MOD_CNT, W	
CALL	MOD	; 인터럽트 발생시 MOD로 진입



	SWAPF	STATUS_TEMP, W	
	MOVWF	STATUS	
	SWAPF	W_TEMP, F	
	SWAPF	W_TEMP, W	
	BCF	INTCON, 2	
	RETFIE		
MOD	;MODE 선택		
	BSF	PORTA,2	;DISP1~4 OFF
	BSF	PORTA,3	
	BSF	PORTB,2	
	BSF	PORTB,1	
	MOVF	MOD_CNT,W	
	SUBLW	.1	
	BTFSC	STATUS,Z	
	GOTO	SEC_DISP	;MOD_CNT가 1인 경우
	MOVF	MOD_CNT,W	
	SUBLW	.3	
	BTFSC	STATUS,Z	
	GOTO	AL_DISP	;MOD_CNT가 3인 경우
	GOTO	HM_DISP	;MOD_CNT가 0인 경우(기본 상태)
HM_DISP	; HOUR/MIN DISPLAY ROUTINE		
	INCF	DISP_CNT	
	MOVF	DISP_CNT,W	
	ANDLW	03H	
	ADDWF	PCL,F	
	GOTO	DISP1	
	GOTO	DISP2	
	GOTO	DISP3	
	GOTO	DISP4	
SEC_DISP	; SEC DISPLAY ROUTINE		
	INCF	SEC_CNT	
	BTFSC	SEC_CNT,0	
	GOTO	S_DISP1	
	GOTO	S_DISP2	

```

AL_DISP      : ALARM DISPLAY ROUTINE
              INCF      DISP_CNT
              MOVF      DISP_CNT,W
              ANDLW     03H
              ADDWF     PCL,F
              GOTO      AL_DISP1
              GOTO      AL_DISP2
              GOTO      AL_DISP3
              GOTO      AL_DISP4

```

-----

```

START_UP
              BSF      STATUS, RP0   ; RAM BANK 1 선택
              MOVLW    B'00000000'   ; PORT I/O 선택
              MOVWF    TRISA
              MOVLW    B'00111000'   ; PORT I/O 선택(SW1,2,3 입력)
              MOVWF    TRISB
              MOVLW    B'00000000'   ; PORT I/O 선택
              MOVWF    TRISC
              MOVLW    B'00000111'
              MOVWF    ADCON1
              ; INTERRUPT 시간 설정 --- 2.048 msec 주기
              MOVLW    B'00000010'   ;2.048 msec
              MOVWF    OPTION_REG
              BCF      STATUS, RP0   ; RAM BANK 0 선택
              BSF      INTCON, 5     ; TIMER INTERRUPT ENABLE
              BSF      INTCON, 7     ; GLOBAL INTERRUPT ENABLE
              GOTO      MAIN

```

-----

;PCL을 사용하는 TABLE 형식은 밑에 넣으면 정상적으로 작동이 안되기 때문에  
;이유) CONV 등이 너무 뒤로 가면 프로그램 카운터가 8bit를 넘어가는 경우가 생겨버림  
;PC의 PCLATH를 수정하기에는 복잡하기 때문에, 이를 해결하기 위해 여기에 위치시킴.

```

CONV
              ANDLW     0FH
              ADDWF     PCL,F
              RETLW     B'11100111'; '0'
              RETLW     B'01100000'; '1'
              RETLW     B'11000011'; '2'
              RETLW     B'11100010'; '3'

```

RETLW	B'01100100'; '4'
RETLW	B'10100110'; '5'
RETLW	B'10100111'; '6'
RETLW	B'11100100'; '7'
RETLW	B'11100111'; '8'
RETLW	B'11100110'; '9'
RETLW	B'11100111'; '0'
RETLW	B'11100111'; '0'
RETLW	B'11100111'; '0'
RETLW	B'11100111'; '0'
RETLW	B'11100111'; '0'
RETLW	B'11100111'; '0'

CONV1

ANDLW	1FH
ADDWF	PCL,F
RETLW	B'00011100';0 DOT OFF
RETLW	B'00011100';1
RETLW	B'00011110';2
RETLW	B'00011110';3
RETLW	B'00011110';4
RETLW	B'00011110';5
RETLW	B'00011110';6
RETLW	B'00011100';7
RETLW	B'00011110';8
RETLW	B'00011110';9
RETLW	B'00011100';0
RETLW	B'00011100';0
RETLW	B'00011100';0
RETLW	B'00011100';0
RETLW	B'00011100';0
RETLW	B'00011100';0
RETLW	B'00011101';0 DOT ON
RETLW	B'00011101';1
RETLW	B'00011111';2
RETLW	B'00011111';3
RETLW	B'00011111';4
RETLW	B'00011111';5
RETLW	B'00011111';6
RETLW	B'00011101';7

```

RETLW      B'00011111':8
RETLW      B'00011111':9
RETLW      B'00011101':0
RETLW      B'00011101':0
RETLW      B'00011101':0
RETLW      B'00011101':0
RETLW      B'00011101':0
RETLW      B'00011101':0
RETLW      B'00011101':0

```

```

DOT_ON      :MODE2(시간 변경)에서 DOT을 깜빡일 때 사용
MOVWF      CHANGE_CNT,W
ANDLW      03H
ADDWF      PCL,F
GOTO       D_ON1
GOTO       D_ON2
GOTO       D_ON3
GOTO       D_ON4

```

```

DOT_OFF     :MODE2에서 DOT을 깜빡일 때 사용
MOVWF      CHANGE_CNT,W
ANDLW      03H
ADDWF      PCL,F
GOTO       D_OFF1
GOTO       D_OFF2
GOTO       D_OFF3
GOTO       D_OFF4

```

```

INC_NUM     :MODE2에서 시간을 증가시키는 역할
CALL       DELAY
MOVWF      CHANGE_CNT,W
ANDLW      03H
ADDWF      PCL,F
GOTO       INC_10HOUR
GOTO       INC_1HOUR
GOTO       INC_10MIN
GOTO       INC_1MIN

```

```

AL_DOT_ON   :MODE3(알람 설정)에서 DOT을 깜빡일 때 사용
MOVWF      ALARM_CNT,W
ANDLW      03H
ADDWF      PCL,F

```

```

GOTO      AL_D_ON1
GOTO      AL_D_ON2
GOTO      AL_D_ON3
GOTO      AL_D_ON4

AL_DOT_OFF  ;MODE3에서 DOT을 깜빡일 때 사용
MOVWF      ALARM_CNT,W
ANDLW      03H
ADDWF      PCL,F
GOTO      AL_D_OFF1
GOTO      AL_D_OFF2
GOTO      AL_D_OFF3
GOTO      AL_D_OFF4

INC_NUM_AL  ;MODE3에서 시간을 증가시키는 역할
CALL       DELAY
MOVWF      ALARM_CNT,W
ANDLW      03H
ADDWF      PCL,F
GOTO      INC_10HOUR_AL
GOTO      INC_1HOUR_AL
GOTO      INC_10MIN_AL
GOTO      INC_1MIN_AL

MOD_CHK     ;MODE를 체크하고 분기시켜주는 역할
CALL       DELAY
INCF       MOD_CNT
MOVWF      MOD_CNT,W
SUBLW      .4
BTFSC      STATUS,Z
CLRWF      MOD_CNT
MOVWF      MOD_CNT,W
ANDLW      03H
ADDWF      PCL,F
GOTO      LOOP      ;MODE0 시/분 시계
GOTO      LOOP      ;MODE1 초 시계
GOTO      CHANGE     ;MODE2 시간 설정
GOTO      AL_LOOP     ;MODE3 알람 설정

```

```

;-----

```

```

MAIN

```

; 변수 초기화

```
CLRF      SEC_CNT
CLRF      DISP_CNT
CLRF      INT_CNT
CLRF      D_10SEC
CLRF      D_1SEC
CLRF      D_10MIN
CLRF      D_1MIN
CLRF      D_10HOUR
CLRF      D_1HOUR
CLRF      D_10SEC_AL
CLRF      D_1SEC_AL
CLRF      D_10MIN_AL
CLRF      D_1MIN_AL
CLRF      D_10HOUR_AL
MOVLW    .6
MOVWF     D_1HOUR_AL    ;기본 알람시간 6시로 설정
BSF       PORTA,4       ;BUZZER OFF
CLRF      MOD_CNT
CLRF      CHANGE_CNT
CLRF      ALARM_CNT
CLRF      TEMP
```

LOOP

```
BTFSS     PORTB,3
GOTO      MOD_CHK
MOVLW     .244          ;1초
SUBWF     INT_CNT, W
BTFSC     STATUS, Z
GOTO      LOOP1
GOTO      LOOP
```

LOOP1

```
MOVLW     0FH           ;하위 4비트만 사용하도록 masking
ANDWF     D_1SEC
ANDWF     D_10SEC
ANDWF     D_1MIN
ANDWF     D_10MIN
ANDWF     D_1HOUR
ANDWF     D_10HOUR
```

CLRF	INT_CNT	
;1초마다 시간 증가 및 단위 설정		
INCF	D_1SEC	
MOVF	D_1SEC,W	
SUBLW	.10	
BTFSC	STATUS,Z	
CLRF	D_1SEC	
BTFSC	STATUS,Z	
INCF	D_10SEC	
MOVF	D_10SEC,W	
SUBLW	.6	
BTFSC	STATUS,Z	
CLRF	D_10SEC	
BTFSC	STATUS,Z	
INCF	D_1MIN	
MOVF	D_1MIN,W	
SUBLW	.10	
BTFSC	STATUS,Z	
CLRF	D_1MIN	
BTFSC	STATUS,Z	
INCF	D_10MIN	
MOVF	D_10MIN,W	
SUBLW	.6	
BTFSC	STATUS,Z	
CLRF	D_10MIN	
BTFSC	STATUS,Z	
INCF	D_1HOUR	
MOVF	D_1HOUR,W	
SUBLW	.4	
BTFSC	STATUS,Z	
CALL	TIME_24	; 24시가 되면 RESET
MOVF	D_1HOUR	
SUBLW	.10	
BTFSS	STATUS,Z	
GOTO	ALARM_CHECK	;알람 체크

```

        CLRF      D_1HOUR
        INCF      D_10HOUR
        GOTO      ALARM_CHECK ;알람 체크

TIME_24      :24시 인지 확인하고 맞으면 시간 초기화(00:00:00)
        MOVF      D_10HOUR,W
        ANDLW     0FH
        SUBLW     .2
        BTFSS     STATUS,Z
        RETURN
        CLRF      D_10SEC
        CLRF      D_1SEC
        CLRF      D_10MIN
        CLRF      D_1MIN
        CLRF      D_10HOUR
        CLRF      D_1HOUR
        RETURN

TIME_24_AL   :MODE3에서 알람설정 시간이 24시 인지 확인하고 맞으면 시간 초기화
        MOVF      D_10HOUR_AL,W
        ANDLW     0FH
        SUBLW     .2
        BTFSS     STATUS,Z
        RETURN
        CLRF      D_10MIN_AL
        CLRF      D_1MIN_AL
        CLRF      D_10HOUR_AL
        CLRF      D_1HOUR_AL
        RETURN

;-----
;MODE0 - 시/분 시계
DISP1
        MOVF      D_10MIN,W
        CALL      CONV
        MOVWF     PORTC
        MOVF      D_10MIN,W
        CALL      CONV1
        MOVWF     PORTA
        BCF       PORTB,2      ;DG3
        RETURN

```



DISP2

```
MOVF    D_1MIN,W
CALL    CONV
MOVWF   PORTC
MOVF    D_1MIN,W
CALL    CONV1
MOVWF   PORTA
BCF     PORTB,1      ;DG4
INCF    INT_CNT
RETURN
```

DISP3

```
MOVF    D_10HOUR,W
CALL    CONV
MOVWF   PORTC
MOVF    D_10HOUR,W
CALL    CONV1
MOVWF   PORTA
BCF     PORTA,3      ;DG1
RETURN
```

DISP4

```
MOVF    D_1HOUR,W
CALL    CONV
MOVWF   PORTC

MOVF    MOD_CNT,W    ;시/분 시계일 때 DOT이 켜져있고
SUBLW   .2           ;시간조정모드일 때 DOT이
                     ;꺼져있는경우

BTFSC   STATUS,Z
GOTO    DISP4_1
MOVF    D_1HOUR,W
IORLW   10H
CALL    CONV1
MOVWF   PORTA
GOTO    DISP4_2
```

DISP4\_1

```
MOVF    D_1HOUR,W
CALL    CONV1
MOVWF   PORTA
```

DISP4\_2

```
BCF     PORTA,2      ;DG2
INCF    INT_CNT
```

```

                                RETURN
;-----
;MODE1 - 초시계
S_DISP1
    MOVF      D_10SEC,W
    CALL      CONV
    MOVWF     PORTC
    MOVF      D_10SEC,W
    CALL      CONV1
    MOVWF     PORTA
    BCF       PORTB,2      ;DG3
    RETURN

S_DISP2
    MOVF      D_1SEC,W
    CALL      CONV
    MOVWF     PORTC
    MOVF      D_1SEC,W
    CALL      CONV1
    MOVWF     PORTA
    BCF       PORTB,1      ;DG4
    INCF      INT_CNT
    RETURN
;-----
;MODE2 - 시/분 시간설정
CHANGE
    BTFSC     INT_CNT, 7    ;0.5초마다 DOT 깜빡임 설정
    GOTO      DOT_ON       ;시간 설정할 위치의 DOT가 깜빡임
    GOTO      DOT_OFF

D_ON1
    MOVLW     10H
    IORWF     D_10HOUR
    GOTO      DOT

D_ON2
    MOVLW     10H
    IORWF     D_1HOUR
    GOTO      DOT

D_ON3
    MOVLW     10H
    IORWF     D_10MIN
    GOTO      DOT

```

D_ON4	MOVLW	10H	
	IORWF	D_1MIN	
	GOTO	DOT	
D_OFF1	MOVLW	0FH	
	ANDWF	D_10HOUR	
	GOTO	DOT	
D_OFF2	MOVLW	0FH	
	ANDWF	D_1HOUR	
	GOTO	DOT	
D_OFF3	MOVLW	0FH	
	ANDWF	D_10MIN	
	GOTO	DOT	
D_OFF4	MOVLW	0FH	
	ANDWF	D_1MIN	
	GOTO	DOT	
DOT	BTFSS	PORTB,3	;SW1 누르면 모드 변경
	GOTO	MOD_CHK	
	BTFSS	PORTB,4	;SW2 누르면 MOV_DIGIT으로
			;이동(자릿수 변경)
	CALL	MOV_DIGIT	
	MOVF	CHANGE_CNT,W	
	SUBLW	.4	
	BTFSC	STATUS,Z	
	CLRF	CHANGE_CNT	
	BTFSS	PORTB,5	;SW3 누를 때 마다 시간이
			;증가하는 INC_NUM로 이동
	GOTO	INC_NUM	
	MOVLW	.244	
	SUBWF	INT_CNT,W	

	BTFSC	STATUS,Z	
	GOTO	INC_1SEC	:시간 변경 중에도 시간이
			:흐르는 것을 반영
	GOTO	CHANGE	
MOV_DIGIT	:시간 변경할 위치 이동		
	INCF	CHANGE_CNT	
	MOVLW	0FH	
	ANDWF	D_1MIN	
	ANDWF	D_10MIN	
	ANDWF	D_1HOUR	
	ANDWF	D_10HOUR	
	CALL	DELAY	
	RETURN		
INC_1MIN			
	INCF	D_1MIN	
	GOTO	INC_CHECK	
INC_10MIN			
	INCF	D_10MIN	
	GOTO	INC_CHECK	
INC_1HOUR			
	INCF	D_1HOUR	
	GOTO	INC_CHECK	
INC_10HOUR			
	INCF	D_10HOUR	
	GOTO	INC_CHECK	
INC_1SEC			
	INCF	D_1SEC	:MOD2 진행 중 시간이
			:흐르는 것을 반영(초)
INC_CHECK	:시간 단위 확인		
	CLRF	INT_CNT	
	MOVF	D_1SEC,W	
	ANDLW	0FH	
	SUBLW	.10	
	BTFSC	STATUS,Z	
	CLRF	D_1SEC	
	BTFSC	STATUS,Z	
	INCF	D_10SEC	

MOVF	D_10SEC,W
ANDLW	0FH
SUBLW	.6
BTFSC	STATUS,Z
CLRF	D_10SEC
BTFSC	STATUS,Z
INCF	D_1MIN

MOVF	D_1MIN,W
ANDLW	0FH
SUBLW	.10
BTFSC	STATUS,Z
CLRF	D_1MIN
BTFSC	STATUS,Z
INCF	D_10MIN

MOVF	D_10MIN,W
ANDLW	0FH
SUBLW	.6
BTFSC	STATUS,Z
CLRF	D_10MIN
BTFSC	STATUS,Z
INCF	D_1HOUR

MOVF	D_1HOUR,W
ANDLW	0FH
SUBLW	.10
BTFSC	STATUS,Z
CLRF	D_1HOUR
BTFSC	STATUS,Z
INCF	D_10HOUR

MOVF	D_1HOUR,W
ANDLW	0FH
SUBLW	.4
BTFSC	STATUS,Z
CALL	TIME_24

MOVF	D_10HOUR,W
ANDLW	0FH

SUBLW	.3
BTFSC	STATUS,Z
CLRF	D_10HOUR
GOTO	CHANGE

;------

:MODE3 - 알람 설정

AL\_LOOP

BTFSC	INT_CNT, 5	:빠르게 DOT 깜빡임 설정
GOTO	AL_DOT_ON	
GOTO	AL_DOT_OFF	

AL\_D\_ON1

MOVLW	10H
IORWF	D_10HOUR_AL
GOTO	AL_DOT

AL\_D\_ON2

MOVLW	10H
IORWF	D_1HOUR_AL
GOTO	AL_DOT

AL\_D\_ON3

MOVLW	10H
IORWF	D_10MIN_AL
GOTO	AL_DOT

AL\_D\_ON4

MOVLW	10H
IORWF	D_1MIN_AL
GOTO	AL_DOT

AL\_D\_OFF1

MOVLW	0FH
ANDWF	D_10HOUR_AL
GOTO	AL_DOT

AL\_D\_OFF2

MOVLW	0FH
ANDWF	D_1HOUR_AL
GOTO	AL_DOT

AL\_D\_OFF3

MOVLW	0FH
ANDWF	D_10MIN_AL

AL_D_OFF4	GOTO	AL_DOT	
	MOVLW	0FH	
	ANDWF	D_1MIN_AL	
	GOTO	AL_DOT	
AL_DOT	BTFSS	PORTB,3	;SW1 누르면 모드 변경
	GOTO	MOD_CHK	
	BTFSS	PORTB,4	;SW2 누르면 MOV_DIGIT_AL으로 ;이동(자릿수 변경)
	CALL	MOV_DIGIT_AL	
	MOVF	ALARM_CNT,W	
	SUBLW	.4	
	BTFSC	STATUS,Z	
	CLRF	ALARM_CNT	
	BTFSS	PORTB,5	
	GOTO	INC_NUM_AL	
	MOVLW	.244	
	SUBWF	INT_CNT,W	
	BTFSC	STATUS,Z	
	GOTO	LOOP1	
	GOTO	AL_LOOP	
MOV_DIGIT_AL	;알람시간 설정할 위치 변경		
	INCF	ALARM_CNT	
	MOVLW	0FH	
	ANDWF	D_1MIN_AL	
	ANDWF	D_10MIN_AL	
	ANDWF	D_1HOUR_AL	
	ANDWF	D_10HOUR_AL	
	CALL	DELAY	
	RETURN		
INC_1MIN_AL	INCF	D_1MIN_AL	
	GOTO	INC_CHECK_AL	
INC_10MIN_AL			

	INCF	D_10MIN_AL
	GOTO	INC_CHECK_AL
INC_1HOUR_AL		
	INCF	D_1HOUR_AL
	GOTO	INC_CHECK_AL
INC_10HOUR_AL		
	INCF	D_10HOUR_AL
	GOTO	INC_CHECK_AL

INC\_CHECK\_AL :시간 단위 확인

CLRF	INT_CNT
MOVF	D_1SEC_AL,W
ANDLW	0FH
SUBLW	.10
BTFSC	STATUS,Z
CLRF	D_1SEC_AL
BTFSC	STATUS,Z
INCF	D_10SEC_AL

MOVF	D_10SEC_AL,W
ANDLW	0FH
SUBLW	.6
BTFSC	STATUS,Z
CLRF	D_10SEC_AL
BTFSC	STATUS,Z
INCF	D_1MIN_AL

MOVF	D_1MIN_AL,W
ANDLW	0FH
SUBLW	.10
BTFSC	STATUS,Z
CLRF	D_1MIN_AL
BTFSC	STATUS,Z
INCF	D_10MIN_AL

MOVF	D_10MIN_AL,W
ANDLW	0FH
SUBLW	.6
BTFSC	STATUS,Z
CLRF	D_10MIN_AL
BTFSC	STATUS,Z



	INCF	D_1HOUR_AL	
	MOVF	D_1HOUR_AL,W	
	ANDLW	0FH	
	SUBLW	.10	
	BTFSC	STATUS,Z	
	CLRF	D_1HOUR_AL	
	BTFSC	STATUS,Z	
	INCF	D_10HOUR_AL	
	MOVF	D_1HOUR_AL,W	
	ANDLW	0FH	
	SUBLW	.4	
	BTFSC	STATUS,Z	
	CALL	TIME_24_AL	
	MOVF	D_10HOUR_AL,W	
	ANDLW	0FH	
	SUBLW	.3	
	BTFSC	STATUS,Z	
	CLRF	D_10HOUR_AL	
	GOTO	AL_LOOP	
AL_DISP1	MOVF	D_10MIN_AL,W	
	CALL	CONV	
	MOVWF	PORTC	
	MOVF	D_10MIN_AL,W	
	CALL	CONV1	
	MOVWF	PORTA	
	BCF	PORTB,2	;DG3
	RETURN		
AL_DISP2	MOVF	D_1MIN_AL,W	
	CALL	CONV	
	MOVWF	PORTC	
	MOVF	D_1MIN_AL,W	
	CALL	CONV1	
	MOVWF	PORTA	
	BCF	PORTB,1	;DG4
	INCF	INT_CNT	

```

        RETURN

AL_DISP3
        MOVF      D_10HOUR_AL,W
        CALL      CONV
        MOVWF     PORTC
        MOVF      D_10HOUR_AL,W
        CALL      CONV1
        MOVWF     PORTA
        BCF       PORTA,3          ;DG1
        RETURN

AL_DISP4
        MOVF      D_1HOUR_AL,W
        CALL      CONV
        MOVWF     PORTC
        MOVF      D_1HOUR_AL,W
        CALL      CONV1
        MOVWF     PORTA
        BCF       PORTA,2          ;DG2
        INCF      INT_CNT
        RETURN

ALARM_CHECK :현재시간과 알람시간 비교해서 확인
        MOVLW     0FH
        ANDWF     D_1MIN_AL
        ANDWF     D_10MIN_AL
        ANDWF     D_1HOUR_AL
        ANDWF     D_10HOUR_AL

        MOVF      D_1SEC,W
        SUBWF     D_1SEC_AL,W
        BTFSS     STATUS,Z
        GOTO      NO              ;1초 단위 알람설정이랑
                                   ;비교 후 같지 않으면 알람X

        MOVF      D_10SEC,W
        SUBWF     D_10SEC_AL,W
        BTFSS     STATUS,Z
        GOTO      NO              ;10초 단위 알람설정이랑
                                   ;비교 후 같지 않으면 알람X

        MOVF      D_1MIN,W
        SUBWF     D_1MIN_AL,W
        BTFSS     STATUS,Z

```

	GOTO	NO	:1분 단위 알람설정이랑 :비교 후 같지 않으면 알람X
	MOVF	D_10MIN,W	
	SUBWF	D_10MIN_AL,W	
	BTFSS	STATUS,Z	
	GOTO	NO	:10분 단위 알람설정이랑 :비교 후 같지 않으면 알람X
	MOVF	D_1HOUR,W	
	SUBWF	D_1HOUR_AL,W	
	BTFSS	STATUS,Z	
	GOTO	NO	:1시간 단위 알람설정이랑 :비교 후 같지 않으면 알람X
	MOVF	D_10HOUR,W	
	SUBWF	D_10HOUR_AL,W	
	BTFSS	STATUS,Z	
	GOTO	NO	:10시간 단위 알람설정이랑 :비교 후 같지 않으면 알람X
	GOTO	YES	:모든 시간 일치 -> 알람O
NO			
	BSF	PORTA,4	:알람 끄기
	MOVF	MOD_CNT,W	
	SUBLW	.3	
	BTFSC	STATUS,Z	
	GOTO	AL_LOOP	
	GOTO	LOOP	
YES			
	BCF	PORTA,4	:알람 켜기
	BTFSC	PORTB,5	:SW3 체크
	GOTO	SUB1	
SUB2	MOVF	MOD_CNT,W	:SW3을 누르면 알람 꺼짐
	SUBLW	.3	
	BTFSC	STATUS,Z	
	GOTO	AL_LOOP	
	GOTO	LOOP	
SUB1	CALL	AL_DELAY	:SW3을 누르지 않는 경우 :6초 지나면 알람 꺼짐
	INCF	TEMP	
	INCF	D_1SEC	
	MOVF	TEMP,W	

	SUBLW	.6
	BTFSS	STATUS,Z
	GOTO	YES
	CLRF	TEMP
	GOTO	SUB2
;-----		
DELAY	MOVLW	.40
	MOVWF	DBUF1
D_LP1	MOVLW	.50
	MOVWF	DBUF2
D_LP2	NOP	
	DECFSZ	DBUF2,F
	GOTO	D_LP2
	DECFSZ	DBUF1,F
	GOTO	D_LP1
	BTFSS	PORTB,3
	GOTO	DELAY
	BTFSS	PORTB,4
	GOTO	DELAY
	BTFSS	PORTB,5
	GOTO	DELAY
	RETURN	
AL_DELAY	:1초 딜레이	
	MOVLW	.10
	MOVWF	DBUF0
D_LP3	MOVLW	.125
	MOVWF	DBUF1
D_LP4	MOVLW	.200
	MOVWF	DBUF2
D_LP5	BCF	PORTA,4
	DECFSZ	DBUF2,F
	GOTO	D_LP5
	DECFSZ	DBUF1,F
	GOTO	D_LP4
	DECFSZ	DBUF0,F
	GOTO	D_LP3
	RETURN	
	END	

## 6. 조원 별 분담 내용

이준영	위세훈
<ul style="list-style-type: none"><li>- 시/분/초 기능 코드 작성</li><li>- 시간설정 기능 코드 작성</li><li>- 알람 기능 코드 작성</li><li>- 결과보고서 작성</li><li>- 발표 PPT 작성 및 발표</li></ul>	<ul style="list-style-type: none"><li>- 시, 분, 초 기능 코드 작성</li><li>- 시간 설정 기능 코드 작성 보조</li><li>- 결과보고서 작성 보조</li><li>- 발표 PPT 작성 및 발표보조</li></ul>

## 7. 결론 및 고찰

마이크로프로세서 설계 실험 수업을 통해 어셈블리어를 접하고, PIC16F876A의 내부 PORT와 여러 가지 기능을 이용하여 LED, 7-seg, interrupt, 통신 등에 대해 공부하였는데, 이제껏 배우고 편하게 쓴 C언어, JAVA와는 다르게 어셈블리어는 직접 주소공간도 고려해야하고 명령어도 복잡하고 프로그래밍 구조가 많이 달라서 초반 코드를 작성할 때 시간이 많이 들었다. 하지만, TERM PROJECT를 계속 하다 보니 어셈블리어 사용에 많이 익숙해지고 마이크로프로세서를 사용하는 방법, 원하는 프로그램을 만들 때 어떠한 요소를 고려해야 하는지 등을 습득하고 공부할 수 있었습니다.

처음에 시작할 때 책에 있는 예제를 바탕으로 시계의 분, 초 기능의 코드를 작성한 후 원하는 기능들을 하나씩 작성해 추가, 실행, 수정 과정을 거쳐 가면서 완성시켰습니다. 중반쯤에 코드에는 이상이 없다고 생각했는데 완전 엉뚱하게 실행되는 문제가 발생하였고 이를 해결하기 위해 시간을 많이 투자했습니다. 여러 가지 코드를 수정해보다 문득 수업시간에 PCL 명령어를 이용하는 경우 PC를 건드리는데 코드가 길어지는 경우 주소 공간에 따라 PCLATH를 수정해줘야 한다는 내용이 기억나서 PCL 명령어를 사용하는 라벨들을 모두 하나의 주소 공간에서 실행되도록 해당 코드를 상단으로 올리는 방식으로 해결했습니다. 그리고 알람 기능에서 부저가 울리도록 설정하니 시간이 카운트되지 않는 문제가 발생하여 알람이 6초간 울리는 동안 시간 증가를 위해 비록 인터럽트에 비해 정확하지는 않지만 딜레이를 이용하여 (AL\_DELAY) 시간이 계산되도록 설정하여 해결했습니다. 이러한 시행착오를 겪으면서 문제가 발생했을 때는 힘들었지만 고민 끝에 해결하며 성취감과 자신감을 얻을 수 있었습니다.

이번 TERM PROJECT를 하면서 우리가 흔하게 볼 수 있고 사용하는 시계를 어셈블리어로 만드는 것이 생각보다 재미있었고 기본적인지만 실생활에 쓸 수 있는 프로그램을 만들었다는 사실에 흥미를 느끼고 뿌듯하였습니다. 다음에는 ARM칩, 여러 가지 모터 등의 여러 가지 자재를 이용하여 좀 더 발전된 결과물을 만들어 보고 싶다는 생각이 들었고 내년 졸업프로젝트를 할 때를 대비한 좋은 경험을 했다고 느꼈습니다.